

**EBOOK**

# Exploring Cybersecurity concepts, risks, OPC UA security and other solutions

TECHNICAL DISCUSSIONS, GUIDES AND IMPLEMENTATION  
TIPS FOR SOFTWARE TOOLBOX PRODUCTS

By Kevin Rutherford

Our mission is to provide you with the right software package to solve your industrial operation challenges.



# Table of Contents

Table of Contents .....	1
Prologue: Knowledge for your important security discussions.....	3
Chapter 1: Difficult Conversations: Open Ports, Business Data Needs, & Cybersecurity Risk .....	4
Start with Listening.....	4
Inside vs Outside the Facility – It’s Still a Risk.....	4
Ports & Attack Surfaces.....	5
Don’t Advanced Firewall Filtering and Intrusion Prevention Services protect us? .....	6
Encryption & Authentication Protects Us, Right? .....	6
But We Are Running a VPN, Isn’t that Safe? .....	7
Storms, and we don’t mean weather, but broadcast storms.....	7
So how does all this relate to OPC UA?.....	7
Chapter 2: OPC UA Security Concepts .....	9
Primary Functions of OPC UA Certificates.....	9
It’s About Trust .....	10
Types of OPC UA Encryption .....	10
How Does Symmetric Encryption Work? .....	10
How Does Asymmetric Encryption Work? .....	12
An Example of How Asymmetric Encryption Works: .....	13
Comparing Symmetric and Asymmetric Encryption.....	14
So How Does Symmetric and Asymmetric Encryption Apply to OPC UA?.....	14
Then what is Sign & Encrypt?.....	15
The Layered Approach to OPC UA Security.....	15
What are the Layers in the OPC UA Security Model?.....	16
Chapter 3: Considerations for Your Own OPC UA Security Architecture.....	18
Options to Do More.....	19
Chapter 4: Alternatives to Opening Inbound Firewall Ports.....	20
Example: Firewall Closed on Production Side, Read-Only .....	21
Example: Adding Read/Write but Keeping Firewall Closed on Data Source side. ....	21
Example: Both Inbound Firewall Ports Closed, using DMZ, Read Only.....	22
Example: Read/Write, Still Keeping Inbound Firewall Ports closed .....	23
Example: MQTT/IoT Security Enhancement.....	23

Chapter 5: OPC UA Security and Specific OPC UA Client and/or Server Solutions .....	25
TOP Server OPC UA Security (Client & Server).....	25
OPC UA Configuration Components in TOP Server .....	26
Connecting Your OPC UA Client to TOP Server.....	35
Connecting TOP Server to another OPC UA Server .....	36
OmniServer OPC UA Security (Server).....	40
OPC UA Configuration Components in OmniServer .....	40
Connecting Your OPC UA Client to OmniServer .....	48
Cogent DataHub OPC UA Security (Client & Server) .....	50
OPC UA Configuration Components in Cogent DataHub.....	51
Connecting Your OPC UA Client to Cogent DataHub.....	65
Connecting Cogent DataHub to Another OPC UA Server.....	67
OPC Data Logger OPC UA Security (Client) .....	69
OPC UA Configuration Components in OPC Data Logger.....	69
Connecting OPC Data Logger to Your OPC UA Server .....	75
OPC Data Client OPC UA Security (Client).....	77
OPC UA Configuration Components in OPC Data Client .....	77
Get Started Building a Custom OPC UA Client with OPC Data Client.....	82
OPC Router OPC UA Security (Client & Server).....	83
OPC UA Configuration Components in OPC Router .....	83
Connecting Your OPC UA Client to OPC Router.....	94
Connecting OPC Router to Another OPC UA Server .....	94
Conclusions.....	95

# Prologue: Knowledge for your important security discussions

Although this e-book focuses on OPC UA, it is impractical to have the discussion about OPC UA security without also considering the broader environment in which OPC UA is used and recognizing that most of our readers come from the operations technology (OT) space. Engineers & technicians in OT are bombarded with automation technologies, tools, and work to do, while your employer's IT & Cybersecurity teams are trying to prevent disastrous security breaches and issues, all while keeping an industrial operation running at full speed.

Ultimately security decisions must be owned by a team that represents the varying interests and concerns in your business. This e-book is not meant to be a cookbook as every industrial user is different and there are great resources from entities like [US NIST who has published guidelines](#) to consider with your IT and Cybersecurity teams.

We will cover 4 topical areas with increasing technical depth, but wrapping up with considerations for you to think about in your applications and systems. You can jump directly to the later sections with the links provided.

1. **Difficult Conversations:** Open Ports, Business Data Needs, and Cybersecurity Risk
2. **OPC UA Security Concepts:** The deep technical concepts - [Certificates & Message Signing](#), [Encryption](#), [Symmetric vs. Asymmetric Keying](#), [Layered Approach of OPC UA](#).
3. **Considerations for Your Own OPC UA Security Architecture:** [Questions](#) to facilitate your discussions and [options to consider on top of OPC UA Security](#).
4. **Alternatives to Opening Inbound Firewall Ports:** What [your alternatives are to opening inbound firewall ports for remote access](#) including examples of situations where this can be needed.

We will then move on to cover the specifics of [configuring OPC UA security for the most commonly used OPC UA capable Software Toolbox client and/or server solutions](#).

# Chapter 1: Difficult Conversations: Open Ports, Business Data Needs, & Cybersecurity Risk

Before we talk about how OPC UA security works, this section is designed to help you prepare for security discussions. We want to start by introducing our partners in the OT space to concepts and IT perspectives we've learned that can facilitate collaborative discussions with IT, towards weighing risks together, evaluating options to address risks, and accomplishing movement of data for operational & business needs.

In the industrial control space, it's common for an engineer to ask IT to open a port on a firewall, router, or other network device to connect between certain systems and/or control devices. We're not just referring to the public internet, but even between network segments such as control and business networks. It's also common for the IT department to have serious concerns about security risks potentially introduced to all of the company's networks by opening a port. This often leads to difficult conversations and requires balancing different priorities and perspectives.

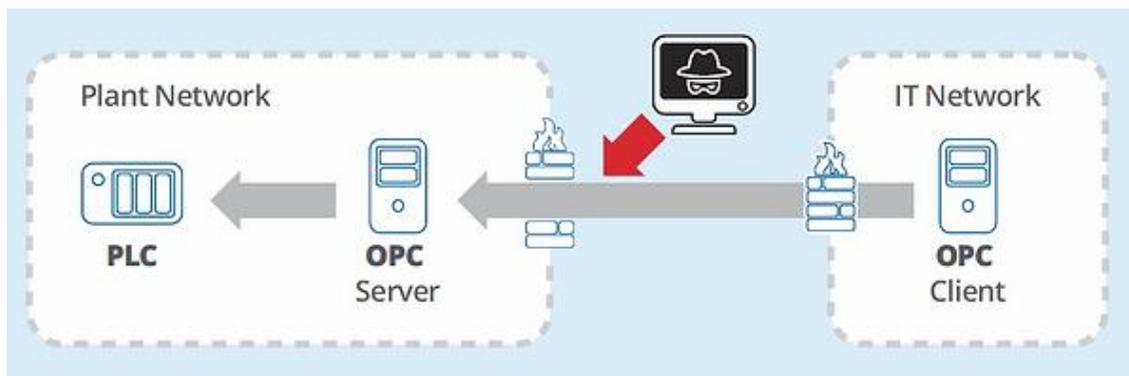
## Start with Listening

Start by listening with the goal to understand the nature of your business, its risk profile, and the concerns people running your business are going to have. If the business is regulated, has access to confidential information from its customers, sells products or provides services that affect health & safety, you can assume it's a target, and that will drive the tone and posture of your cybersecurity team's concerns.

## Inside vs Outside the Facility – It's Still a Risk

The topic of the "inside job" where an employee or contractor is physically in the building and has a computer that can directly join the network is the reason why IT will be concerned about any open port between business and production networks or other network segments.

Their job is to protect all, so even if the firewall to the internet is totally closed to inbound traffic, they think about the "what if". Several well-known attacks on Industrial Control System (ICS) networks originated through a rogue device connected to a trusted network inside the facility.



Also, well-known email link, malware email attachments and other “phishing” schemes create the risk of a user on the production or business side “inviting an attacker in”. That’s why those attacks are so popular and, once on the network, that attacker may as well be physically present, depending on how your network is setup.

## Ports & Attack Surfaces

Ports are doors to your systems and network. Any inbound port open to traffic from the public internet is a huge risk. But those same ports open between internal networks also pose risks.

When a port is closed on a device, security is in the hands of the device to not accept any requests to open a connection for TCP traffic and reject UDP connectionless traffic to those closed ports. Attackers will receive no response when probing for vulnerabilities.



When you open any port with no restrictions you have opened a door completely and told the firewall to let all traffic flow, bypassing methods, algorithms, & technology in the purpose-built security device. An open port is an attack surface. Even one port is a place for someone to probe.

Then who is in charge of security after that? It’s the software applications that the traffic flows to that you are now depending on for that security. Those applications are nearly always built to perform a specific function and are not specialized and hardened for network security like a firewall. Attackers probe for open ports, exploit known software weaknesses, and look for weaknesses in the software such as a factory default password, an easily guessed password, an injection attack, or other vulnerability.

If you have no other options and must open a port, even on your internal network, your IT team will likely insist on limiting access to known IP addresses and employ a whole host of other measures to manage the risk, which may cause you to still consider other options we mention later.

But even encryption, authentication, VPNs, and other mitigating technologies sometimes mentioned as “the solution”, have their own risks, which is why your IT team may be taking a hard line. Let’s look at those risks.

## Don't Advanced Firewall Filtering and Intrusion Prevention Services protect us?



Today's enterprise-class firewalls are very capable of inspecting traffic, identifying suspicious or known attacks, and blocking them even on an open port. There are even specialized firewalls that can inspect industrial control protocol traffic and restrict it by protocol function code. You can also set up rules that say "only allow traffic from these IP addresses, or even in some cases these MAC addresses" which can make the attack surface smaller but never totally eliminate it.

However, attackers are smart too and always adapting. Also, remember a human configures and maintains those filters. One mistake and the attack surface is enlarged, or critical data flow is cut off. Some firewalls can run a proxy on an open port and the proxy inspects the traffic using rules, algorithms, and detection mechanisms that are constantly updated by the firewall device vendor, or your cybersecurity team, or both. When using those technologies, it's important to realize you are trusting that device's algorithms and any rules that were defined by humans managing the firewall. There is still an attack surface.

## Encryption & Authentication Protects Us, Right?

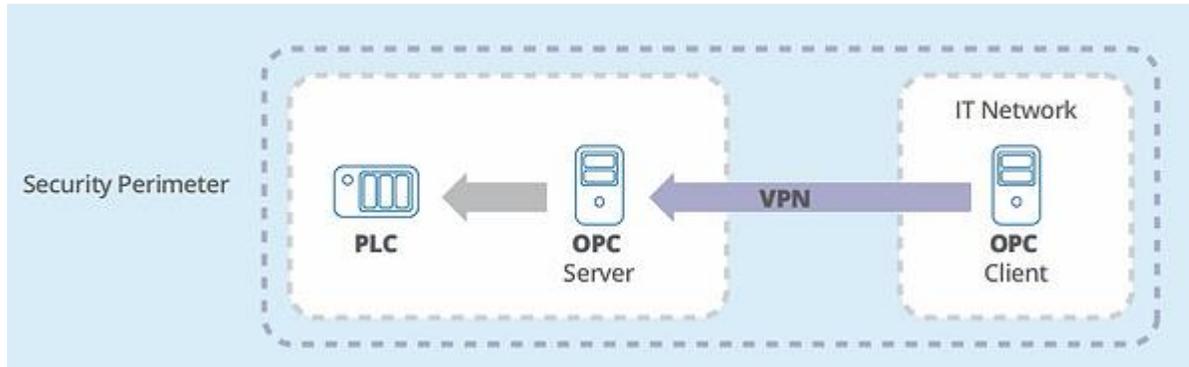
We will get into [how OPC UA uses encryption](#) in the next section. Encryption keeps someone from reading your data if they can't decode it. It does not prevent attacks. Encryption also means that any filtering in your firewall can't read the data so they cannot determine if an exploit is included in the data, so the firewall packet inspection is actually subverted by encryption.

There are well-publicized break-ins that were performed by attacking the encryption mechanisms on devices, server operating systems, and software applications. If you've had to deal with IT shutting off the older TLS1.0 and TLS1.1 encryption support on your systems, that's why as those encryption methods are well known attack surfaces.

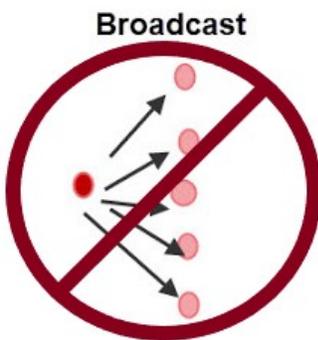
Similarly, authentication is just another thing for an attacker to try and exploit. Even if they don't get in, they can overload your software application with requests to the point the application is unable to perform its main purpose and is spending all its time fighting off requests, at best.

## But We Are Running a VPN, Isn't that Safe?

VPN's sound safe as a secure way of allowing traffic to flow into or within the secure environment of the plant. But, actually, it's the other way around, especially with remote VPN users. Using a VPN simply expands the security perimeter of the plant or network segment to include the remote VPN client and anything it has been exposed to or is connected to. This is how the NSA EternalBlue exploit propagated the well-known WannaCry virus, and still remains a serious threat today. VPN's are powerful but are limited by the security of the connected device(s).



## Storms, and we don't mean weather, but broadcast storms



Another concern IT may express about industrial control protocol or devices you add to the network is whether or not they generate broadcast traffic on the network.

Broadcast traffic is sent from one device to all devices on the network. It's like someone walking into a quiet office of work desks and screaming – no one can hear, think or talk over the screaming, so broadcast traffic is detrimental to overall network performance at best. Broadcast traffic can take a network to its knees, feel like an inside attack, and could stop production.

## So how does all this relate to OPC UA?

As the most current of the OPC standards that includes encryption and other security capabilities, OPC UA technology adoption continues to grow in the industrial control space. Even with that security, asking your IT department to open a port to connect an OPC UA client to an OPC UA server will almost certainly result in scrutiny.

While OPC UA does only require a single open port which, compared to DCOM technologies used in OPC Classic, is less risky, it's still an open port. The same would apply to any protocol that requires a port to be opened. There are realistic alternatives that work with OPC UA and other popular protocols like MQTT, requiring no open inbound ports which we'll cover next.

On the subject of broadcast traffic, OPC UA uses a TCP port, which is socket-based, as opposed to UDP. UDP, which is connectionless, is typically used for broadcast messages where a connection to the receiving host/hosts is not required first (for example, the SNMP protocol commonly uses UDP). Because OPC UA uses TCP for communications that means it will not generate broadcast traffic. If you're interested to learn more about OPC in general, consider our [OPC Getting Started FAQs Guide](#).

## Chapter 2: OPC UA Security Concepts

In this section, the goal is for you to learn more about OPC UA Security, the terminology, and technologies used to implement OPC UA Security.

The use of certificates in cryptographic applications and online communication protocols is nothing new and can practically be traced all the way back to the 1970s when the ‘framework’ for public key encryption came into being. With the demands of digital transformation, remote access to data, and IoT/IIoT, we are seeing more and more systems – that have traditionally been air-gapped and kept offline – being brought onto company networks.

First we need to look at what OPC UA Certificates are and what they provide, how they are used in OPC UA, and how they fit into the security ‘stack’ of OPC UA. Then we will examine OPC UA encryption methodology and the key concepts of its [layered security model](#).

### Primary Functions of OPC UA Certificates

If you have ever configured an OPC UA Connection between an OPC UA Client and OPC UA Server – you are probably familiar with OPC UA Certificates. OPC UA makes use of the X.509 certificate standard, which defines a standard public key format and is used in OPC UA for three primary functions:

1. **OPC UA Message Signing Validates Communications Integrity**

The application uses its private key to generate a message signature/hash, which can then be validated by using the corresponding public key certificate. If the private key signature checks out, the message is guaranteed to come from the corresponding application. Modifying the message in transit (as might be the case in a man-in-the-middle-attack) would result in the message signature/hash no longer being correct – showing that the message was modified in-flight.

2. **OPC UA Message Encryption Keeps Communications Safe From Prying Eyes**

In the same way that an application’s private key can be used to sign a message to guarantee it was generated by the approved application, the public key can be used to encrypt messages. Once a public key is used to encrypt a message, only the application with the corresponding private key can decrypt it. An attacker could even have a copy of the public certificate and they would not be able to decrypt the message that was just encrypted; a public key is used for encryption only – it cannot be used to decrypt its own messages.

### 3. OPC UA Application Identification Provides Measure of Trustworthiness

Naturally being able to sign messages and encrypt/decrypt them would not be much good to us if there was no way to determine whose certificate we were working with. Each OPC UA certificate, therefore, also provides identifying information on what application generated the certificate, when it was generated, by who it was generated, what the certificate can be used for, how long the certificate should be valid for, where it was generated, and many other things.

#### It's About Trust

Central to those three functions comes the concept of trust. When two OPC UA capable applications first connect, they exchange their Public Keys (also known as their application certificates or just OPC UA Certificates), while keeping their corresponding Private Key...well...private.

It is now up to the user to go into each application – the client and server – and trust the other side's public certificate. But be careful – trusting a certificate in your application may seem like a trivial step, but you are telling your application that the other application is trustworthy and allowed to connect/communicate. Make sure to validate that the certificate you are trusting has been vetted – after all, a locked door is no good when you give the bad guy the key.

Understanding what OPC UA Certificates are used for is one thing but understanding *how* they are used is the next step.

#### Types of OPC UA Encryption

Now that we've had a look at the functions OPC UA Certificates serve in the context of OPC UA security, we need to consider what happens to messages after you have trusted the OPC UA certificates and have enabled security on the OPC UA endpoint. An OPC UA endpoint is the address that you use to access an OPC UA Server.

Specifically, we will need to discuss what does “Sign & Encrypt” mean on an OPC UA endpoint configuration and how can we be sure that the data is truly secure? We will get to that but first we need to explore some more technical terminology so that answer will make sense.

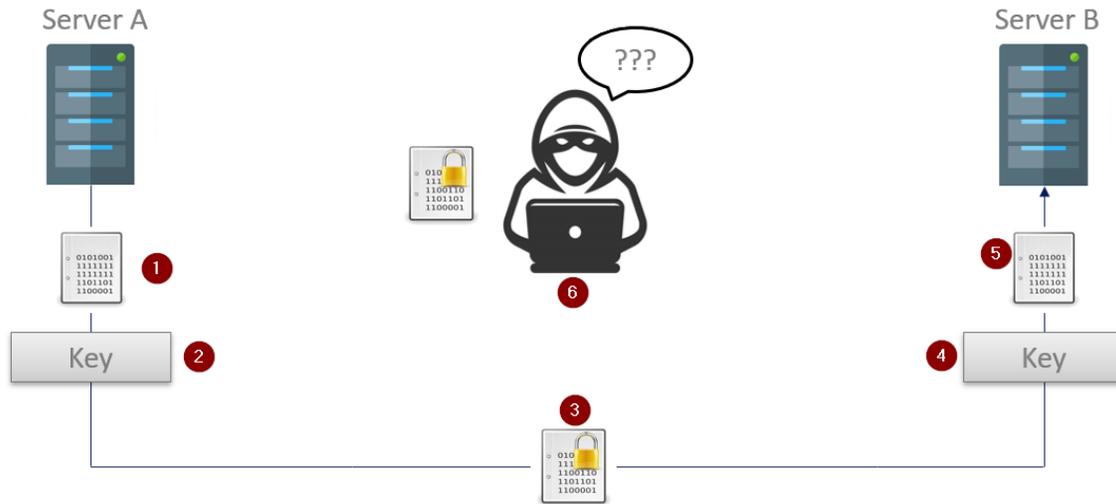
To really understand how OPC UA Security between two applications works, we need to know a little about how symmetric and asymmetric (or public key) encryptions work – because at the end of the day they are not so different from one another. They are both used in OPC UA communications security to achieve security while also meeting performance needs. Let's learn how.

#### How Does Symmetric Encryption Work?

Symmetric encryption is so named (quite aptly) because the same key is used to both encrypt and decrypt the encoded message, not unlike a physical lock where the same key is used to lock

and unlock the lock. This symmetry is great because it is a very fast way to encrypt/decrypt information because the same key is used for both, and because it is reasonably secure, although not as secure as its asymmetric counterpart.

### An Example of How Symmetric Encryption Works:



1. Server A (the OPC UA Client in this case) has an unencrypted message that it wants to send to the OPC UA Server on Server B.
2. Server A uses the previously exchanged symmetric key to encrypt the message
3. The encrypted message – safe from prying eyes – is now sent to the OPC UA Server running on Server B.
4. Where the same key that was used to encrypt the message is now used to decrypt it
5. And the original message is then delivered to the OPC UA Server
6. Any *unapproved* parties who do not have the key would end up with an encrypted message they can do nothing with

If the OPC UA Server running on Server B wanted to send an encrypted response it would simply need to use the same – previously exchanged – key to encrypt the message, and the process would repeat in reverse.

Because the same key is used for encryption and decryption, Symmetric Encryption is fast – and therein lies its primary benefit over Asymmetric encryption - speed. But that speed comes at a price, and that is security.

The example only briefly touched on arguably the most critical step of symmetric encryption, and that is the key exchange. How do you securely perform an exchange that protects the key in a way where malicious parties cannot get access to it, and where the key can be easily exchanged with a new one if the old one were ever to be compromised?

And because, again, the same key is used both for encryption and decryption, *IF* a malicious party were to ever get ahold of the key – it would be able to not only decrypt messages from both

sides, but would be able to generate its own messages that could now potentially be treated as valid.

Another less than ideal aspect of symmetric encryption is that the number of certificates can also be prohibitive when dealing with more than just two machines. With a single client/server pair, there is only one key needed, but with 100 servers/clients – where each machine might want to talk to each of the other machines in the system – the number of needed certificates grows exponentially. Asymmetric encryption addresses many of the concerns of symmetric encryption.

## How Does Asymmetric Encryption Work?

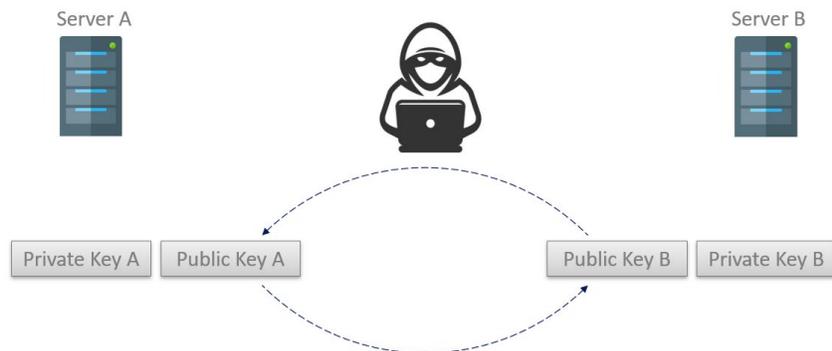
Asymmetric encryption – or public key encryption – (also quite aptly named) is considered asymmetric because the same key is no longer used for encryption and decryption. How is that possible you ask? How can one key be used to encrypt data, and another key be used to decrypt it?

Basically, with asymmetric encryption, each application will have two certificates that are mathematically linked (more on what this means in a second):

1. A private certificate that is used for decryption and message signing
2. A public certificate that is used for encryption.

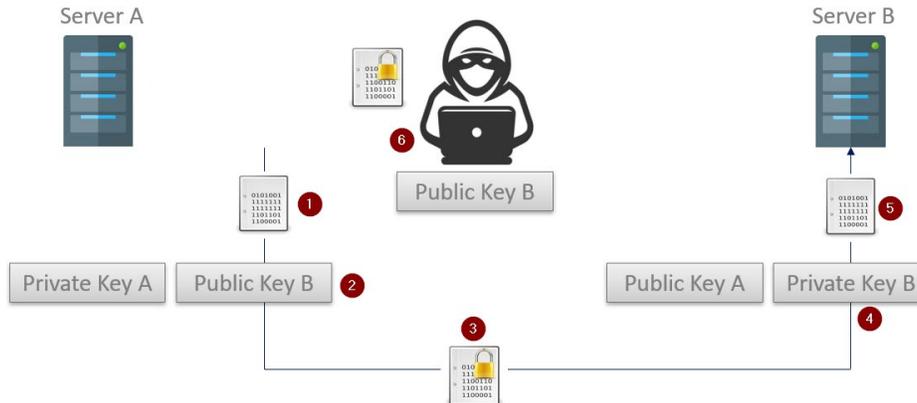
These keys are mathematically linked meaning that a message that is encrypted with a public key can ONLY be decrypted with the corresponding private key.

The first step with any connection that will use asymmetric encryption is that the public keys are exchanged.



Because an encrypted message can only be decrypted using the corresponding private key, applications can be very liberal in who they give their public keys to since anyone that intercepts an encrypted message cannot decrypt the message (even if they have a copy of the public key used to do the encryption). This also means an application only needs a single public key that can be used by many applications – solving the certificate count issue we see when using symmetric encryption.

## An Example of How Asymmetric Encryption Works:



1. The OPC UA Client running on Server A starts with an unencrypted message that will be sent to the OPC UA Server running on Server B. The public keys have already been exchanged.
2. The UA Client will use the public key that was received from the OPC UA Server to encrypt the messages and...
3. The encrypted message is sent off (where it is intercepted by a malicious third party)
4. The OPC UA Server on Server B uses its private key to decrypt the encoded message...
5. Which is then processed.
6. The bad actor that intercepted message as well as Server B's public key is still unable to decrypt the captured message.

There is another great feature that this mathematical link between the private and public key brings us; not only can the public key be used to encrypt traffic that only the corresponding private key can decrypt. The server can also use the private key to *sign* a message and generate a message hash that – when compared against the public key – guarantees the message was not altered, and that the message originated from the machine we think it did.

Earlier I said that the public and private keys are mathematically linked, and I want to revisit that. What this means is that one key (or a part of one) was generated using the other in some way, or that both keys were generated from the same large, random, prime number.

How that generation is accomplished depends on the algorithm used (RSA, Elliptic Curve, etc.) but the important piece is that there is a mathematical relationship between the two because of that operation. It is nearly impossible to derive the public key by just knowing the private key, and nearly impossible to derive the private key just knowing the public key. We say “Nearly impossible” because, given enough time and enough computing power and enough mathematical brain power, nothing is completely unbreakable.

While significantly more secure than symmetric encryption, asymmetric encryption has one major drawback and that is speed. While these mathematically linked keys are fantastic, having

to do that math on every message can add a significant amount of computational overhead. And that brings us to our comparison: which one is better?

## Comparing Symmetric and Asymmetric Encryption

Symmetric Encryption		Asymmetric Encryption	
Pros	Cons	Pros	Cons
Incredibly Secure	Same key used for encryption and decryption (compromised keys are high impact)	Even More Secure than Symmetric	Slower than symmetric encryption (by a non-trivial margin)
Encryption and decryption are faster	No good method for securely exchanging keys	Public key can be shared freely (as long as the private key stays secure)	
No Complex Math Relationships		Compromised key would only impact one direction of communications.	
		Bad actors intercepting a public key is low impact	

Ultimately, to answer the question of which type of encryption is better, you will need to consider the objectives for your particular project. If the highest level of security is of the utmost importance regardless of any other factors, asymmetric encryption is obviously the right option. If your project can accept a slightly less secure level of encryption in exchange for significantly faster performance and less overhead, symmetric encryption is likely the right option.

## So How Does Symmetric and Asymmetric Encryption Apply to OPC UA?

When you exchange the application/OPC UA certificates for your client and server, you are exchanging the public key as part of those certificates. This means that when you are choosing to encrypt your OPC UA connection, OPC UA uses asymmetric encryption to secure the initial connection, but – in order to work around the slower communication performance with asymmetric encryption – once the channel is secure a symmetric encryption key is exchanged for communications.

This voids the two primary concerns with both encryption standards:

1. Since communications are performed using symmetric encryption, we gain the speed benefit offered there.
2. Since the connection uses an asymmetrically encrypted connection when the symmetric key is exchanged, this significantly reduces the risk of the key being intercepted and used to break communications.

To keep the connection secure for the long term, the “Secure Channel” (more on this concept shortly as we dive into the layers of the OPC UA Security model) is regularly and automatically renewed by the OPC UA Client and Server, so the same symmetric key is not used long term, which removes another of the symmetric key risks.

## Then what is Sign & Encrypt?

So when an OPC UA Endpoint talks about Sign vs Sign & Encrypt, it is quite literally talking about how the public and private keys will be used.

Signing is proving who you say you are and uses Asymmetrical keying in that the message is signed by your private key, and can only be decrypted using your public key.

Encrypting is about protecting the data so only the desired receiver can read it. Encryption of the actual data message in OPC UA is done using symmetric keying for speed; HOWEVER, the secret symmetric key is exchanged over the channel that was secured using Asymmetrical keying.

A Signed endpoint lets the OPC UA Client and Server authenticate who they are but then the data is exchanged in an insecure fashion. We’re not sure why you would do this but it’s an option.

Sign and Encrypt gives the best security and balances the performance/security interests.

## The Layered Approach to OPC UA Security

So far, we have taken a very general look at OPC UA Certificates and how they are used by OPC UA Clients and OPC UA Servers to keep data secure. It’s time to take a step back and look at OPC UA Security in general.

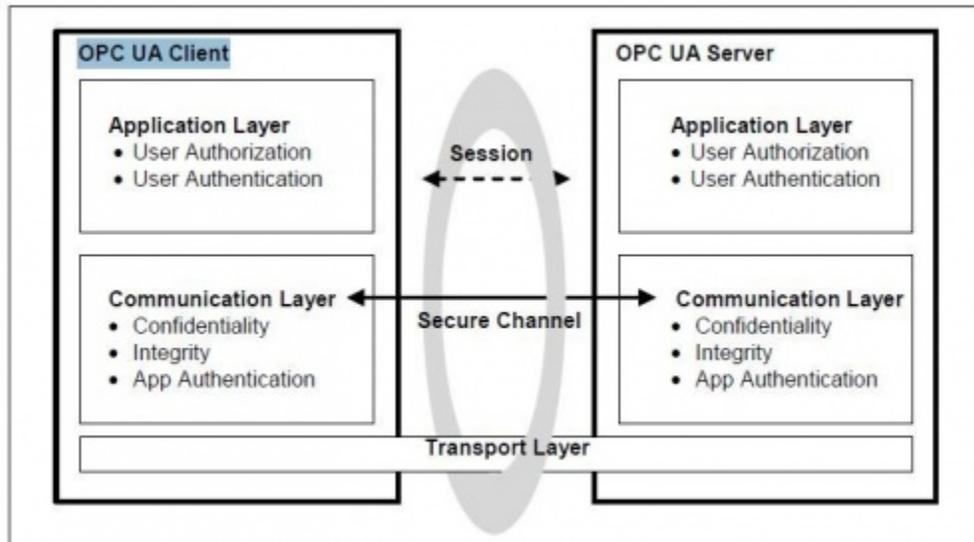
A core pillar of OPC UA is the focus on security, not just for data integrity reasons, but also for service availability. The OPC UA Specs summarize the security focus in three areas:

1. Authentication between client and server applications
2. The ability to determine whether a user is authorized to connect and/or perform the requested action
3. The confidentiality and integrity of the communications.

This means that a strictly layered approach to security is pivotal to an OPC UA Implementation, where each layer is responsible for verifying that the connection/action is allowed, and any unapproved actions can be rejected quickly.

## What are the Layers in the OPC UA Security Model?

The OPC UA Spec documentation visualizes the OPC UA security model as having three layers:



1. **OPC UA Transport Layer** – This is the lowest layer, and the first line of defense. Here we are concerned about the IP address of the machine and the port on which the application is listening. (In most cases, relying on an IP address or port remaining unknown is not really security, just a security incident waiting to happen).

This layer can also include any defenses outside the scope of OPC UA (e.g. firewalls, access control lists, etc.) that could reject a connection before it is ever established.

2. **OPC UA Communication Layer** – This is where most of the activity occurs. When the OPC UA Client connects to the OPC UA Server, a Secure Channel is established where the certificate exchange occurs. This certificate is then used to not only authenticate the applications and hosts making the connections but also encrypt and sign the messages being sent.

If the certificates that the client and/or server are using are not trusted – then the OPC UA Application can reject the connection attempt as the Secure Channel is being established. This is significant because insecure connection attempts should be rejected as low on the protocol stack as possible – to avoid denial of service or resource exhaustion type of attacks; where a malicious app simply opens connections to consume server-side resources and drive the server to a point where it is unable to service legitimate connection attempts.

3. **OPC UA Application Layer** – this where user authentication and OPC UA call/command authentication occurs. By the time we make it to this layer, we already know that the host and application making the call is trusted, the conversation between OPC UA Client and OPC UA Server is secured and, as such, the only thing left to verify is whether the user interacting with the application is authorized to access the resources in question.

The way this typically works is that the user credentials are provided when the session is activated, and if the user is authorized to activate the session (and connect to the UA Server) then the UA Server will return a security ‘token’ that all future calls made by this user must include. By including this ‘token’ on future calls, the server can reject access to specific resource (i.e. cert tags/nodes may not be accessible by every user, some users might have read only access, etc.) Our technical blog has an [example of user security configuration](#).

## Chapter 3: Considerations for Your Own OPC UA Security Architecture

With those three security layers in mind, there are several considerations that should be kept in mind when designing a system that implements OPC UA Clients and Servers.

1. Do not rely on security through obscurity. If the only security implemented on the system is the hope that an attacker cannot figure out the server IP and port, you are a short [NMAP scan](#) away from a potentially compromised system. This is especially dangerous on publicly exposed OPC UA Servers or any system.
2. Be conscious of what certificates you are trusting. Given the importance of the Secure Channel and the critical nature of the proper use of the OPC UA Application Certificates, simply accepting and trusting every certificate effectively negates the purpose of having this layer. (i.e. a firewall is not effective if it simply allows every connection) Take care and review certificates before trusting them; things to look for are:
  - a. Is the host one that I trust and want to allow to connect to me?
  - b. Is the application one that I trust and want to allow to connect to me?
  - c. How long is this certificate valid for and does that adhere to my IT best practices to make sure certificates are regularly updated and renewed?
3. All this talk of certificates becomes somewhat pointless if the OPC UA Server endpoint is not configured to use security. You may have good business reasons to not use OPC UA security, and ultimately you own that decision, but it is something that must be taken into consideration when designing a system.
  - a. Is the system going to be exposed publicly or only be accessible on an intranet?
  - b. Is the network going to be accessible by parties that – while authorized to be on the network – should not be able to monitor the OPC UA Communications?
  - c. How many clients will be connecting on this system?
  - d. Are we using any other security (like user authorization or firewalls)?
4. Does my server allow me to configure users?
  - a. And, if yes – what level of granularity does my OPC UA Server support for allowing me to configure user access permissions (i.e. can I restrict read/write access, access to specific tags, etc.)?
5. Are we siloing user access to only those resources the user needs to access? Or is every user a root user with full administrative rights?

The list of considerations to evaluate when planning the design for a secure OPC UA architecture is considerably longer than what is listed above, but these offer a starting point to

drive a conversation in your organization to ensure a system that meets the unique requirements of your project.

## Options to Do More

Ultimately decisions around opening ports, security levels and related matters must be made by you and your IT team in the context of your business needs and cybersecurity stance. We are not in the business of rendering advice on whether to open ports or not, or how to do that within your security stance but, as mentioned earlier, there are experts that can help, such as your company's cybersecurity team and the US NIST who has [published guidelines](#) to consider.

## Chapter 4: Alternatives to Opening Inbound Firewall Ports

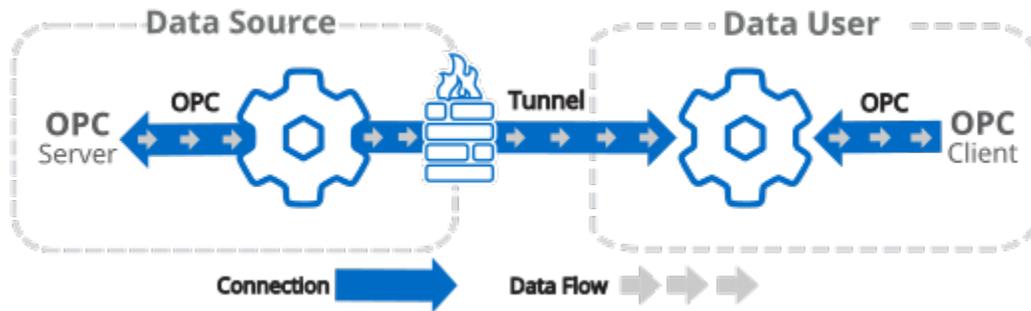
For in-plant connections across routers or firewalls, the [Cogent DataHub](#) and its secure tunneling protocol enable you to do many things to move data, including OPC Classic (DA, A&E) and OPC UA data, such as create read only one-way channels, or separate read and write channels into their own secure paths, and even allow data in without opening an inbound open port. For OPC DA users, DCOM is eliminated from the connectivity stack.

So why would you do this with OPC UA? There are several reasons

1. **Added Security to Keep Inbound Firewall Ports Closed** – as strong as OPC UA security is, it requires an inbound open firewall port. If that is a concern, as you'll see in the examples here, DataHub can working with a DMZ can eliminate the requirement for any inbound firewall ports on the production or business side or both. You'll still have OPC UA security place from the UA client to DataHub and UA server to DataHub, and the enhanced, advanced security of the DataHub connections
2. **Performance** – the DataHub Tunneling protocol described in the first example, is significantly more lightweight, using less bandwidth, for applications where that matters. With the DataHub being capable of processing 50,000 data changes a second at very low latency, applications can benefit.
3. **Resiliency** – when DataHub's are talking a heartbeat is shared at a user definable time and retry period. If there is a loss in network connectivity, OPC data quality flags are set based on user configuration, and statistics tags that can be monitored share the health of the connection. Connections are automatically restored at the speed you choose to setup your heartbeat and retry.
4. **More Data Choices than OPC** – the same connection can move Modbus, spreadsheet, database, MQTT/IoT data, and even USB or IP camera data.

Let's look at some examples that go beyond the traditional client application having to have access through a firewall to access production data. Later in this document we provide [details on implementing OPC UA security with DataHub](#).

## Example: Firewall Closed on Production Side, Read-Only



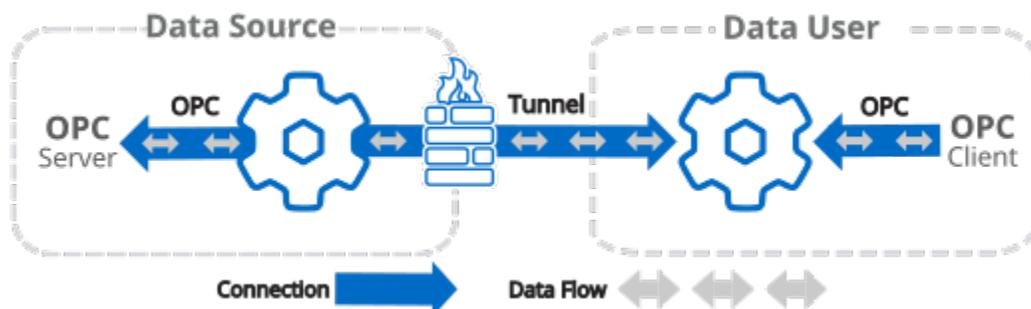
In this case, the DataHub on the plant side initiates the connection on a user definable TCP/IP port and pushes data up to the DataHub on the client side. There are no VPN's in use here. The DataHubs communicate using the DataHub Tunneling Protocol (DHTP) that operates at the application level in the TCP/IP stack if you're curious, leveraging the standard Windows TCP/IP stack.

We always recommend users use the ability to use SSL/TLS, their own certificates, and username/password authentication over the tunnels for best results.

The OPC client and servers can be OPC DA, A&E, UA, or UA A&C servers and clients. In our examples we are sharing, remember the data can be from any [DataHub source](#), not just OPC.

The OPC client connects to that DataHub instance and thinks it's connected to a local OPC client. The data from the production OPC server is automatically mirrored up to the client side DataHub. When there is a data change, the client gets the value, quality, & timestamp. Writes from the client will not be passed down, and the client side cannot change that. The production side oversees their own security choices.

## Example: Adding Read/Write but Keeping Firewall Closed on Data Source side.

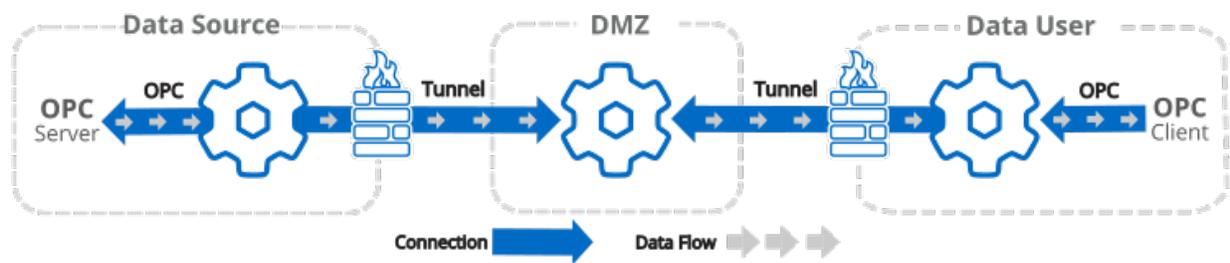


If the production side wants to allow writes, they can keep the inbound firewall ports closed at their Data Source side in the drawing above. They turn on a setting in the DataHub on the Data

Source side of this diagram, to allow writes to come back down from the client over the same socket. Think about when you browse the internet. Your firewall let's HTTPS port 443 secure traffic out and does not have an open port for just anyone on that same port, but allows answers to it's requests to come back. DataHub is doing something like that, so that you can keep that inbound port closed.

If they want to tighten things up, they can setup multiple connections and assign data groups (also known as data domains) to different connections and have the writes separated from the reads or a read only connection and a read-write connection.

### Example: Both Inbound Firewall Ports Closed, using DMZ, Read Only

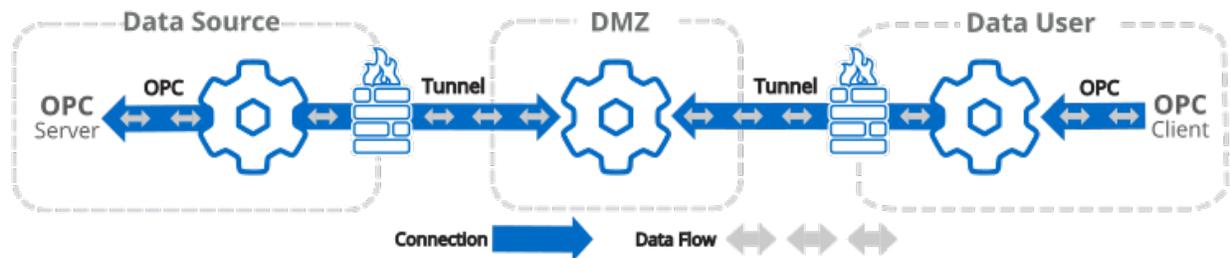


In this architecture, we add a DMZ server in between the Data User and the Data Source. We see this a lot in company architectures between business and production.

Like other scenarios, there are no inbound ports open on the Data Source and Data User firewalls. The DMZ server will have to have inbound ports open to receive data from the Data Source and Data User, but that is customary and expected with a security hardened DMZ server.

Since this is a read only scenario, the DataHub at the Data Source is configured to push data changes from its sources such as the OPC server shown, up to the DMZ DataHub through an outbound only port. The Data User has connected to DataHub in the DMZ to receive any data changes that occur as they come in from the Data Source DataHub. The DataHub's handle all this in milliseconds, only slightly adding to the overall network transport time, while significantly increasing security.

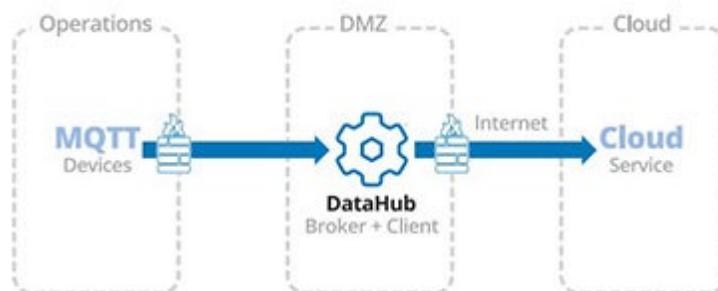
## Example: Read/Write, Still Keeping Inbound Firewall Ports closed



Turning the last example into a bi-directional data flow is simply a matter of making one setting at the Data Source and Data User DataHub instances, and the data can flow both ways, and still the firewall ports are closed at the source and user sides. This is simply an extension of the earlier example where we explained how DataHub allows this to work.

## Example: MQTT/IoT Security Enhancement

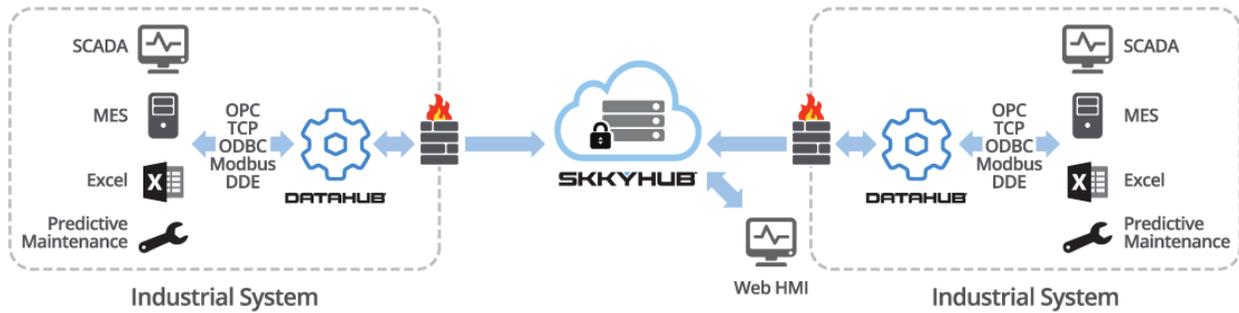
What if you have MQTT devices in the production environment and want to push data to the cloud. It's common to think that's perfectly safe. Doing this would require your secure, protected production network to have direct, unrestricted access to the internet which can present a serious security problem and many cybersecurity authorities frown upon that architecture.



The solution is to run the data through a DMZ, which adds to the complexity. [Learn more about how DataHub helps MQTT users online.](#)

## Cloud Based Transfers

For remote monitoring and outside of the plant applications, the Cogent DataHub and SkkyHub service can help you move or share data securely by “reversing the connection” so that there are no inbound ports open. [Read more about options for your own remote access requirements](#)



# Chapter 5: OPC UA Security and Specific OPC UA Client and/or Server Solutions

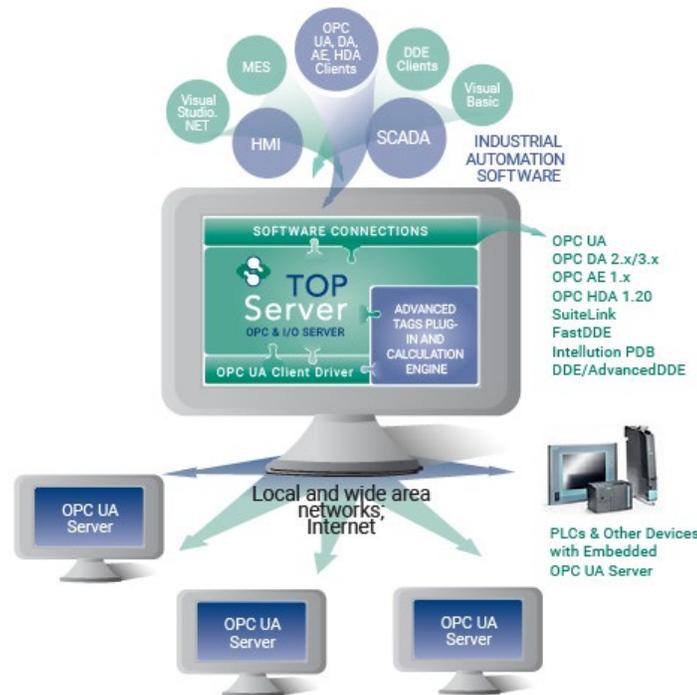
Now that we've covered the technology and considerations behind OPC UA client and server security, the remainder of this e-book will cover the OPC UA settings and configuration for our most frequently used OPC UA client and server software solutions at Software Toolbox.

The details and steps provided, along with references and links to other resources for each solution should make it easy and straightforward to get your OPC UA clients and servers connected to Software Toolbox solutions.

## TOP Server OPC UA Security (Client & Server)

The first thing to know about TOP Server with respect to OPC UA is that it natively exposes an OPC UA server interface for access by OPC UA client applications such as HMI, SCADA, MES, Historians and other applications that need to access process data from PLCs, RTUs and other automation control devices. That OPC UA server interface is available at no additional cost for any driver or plug-in.

With the addition of a licensed [OPC Client Suite for TOP Server](#), the OPC UA Client driver enables TOP Server to also be an OPC UA client to other OPC UA servers (and even other TOP Server installations – an architecture we refer to as OPC UA tunneling). This allows users with client applications that do not currently support OPC UA to still access OPC UA servers (i.e. OPC DA, SuiteLink and DDE applications).



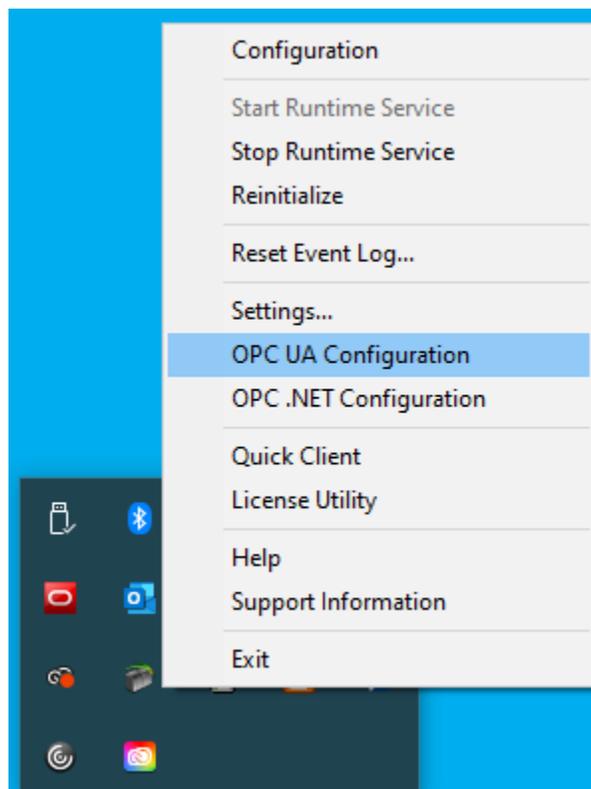
## OPC UA Configuration Components in TOP Server

The TOP Server settings relevant to OPC UA (both client and server) can be found in several different locations of the TOP Server's user interfaces:

### 1. *TOP Server OPC UA Configuration Manager*

Accessible by right-clicking on the TOP Server Administration icon in the Windows system tray and selecting "OPC UA Configuration" from the menu, the OPC UA Configuration Manager for TOP Server is where the bulk of the connection-specific and security-specific settings for OPC UA are configured.

This includes defining OPC UA endpoints, the encryption supported for those endpoints, OPC UA security certificate management for both the OPC UA server interface and OPC UA Client driver and more.



The following sections are available:

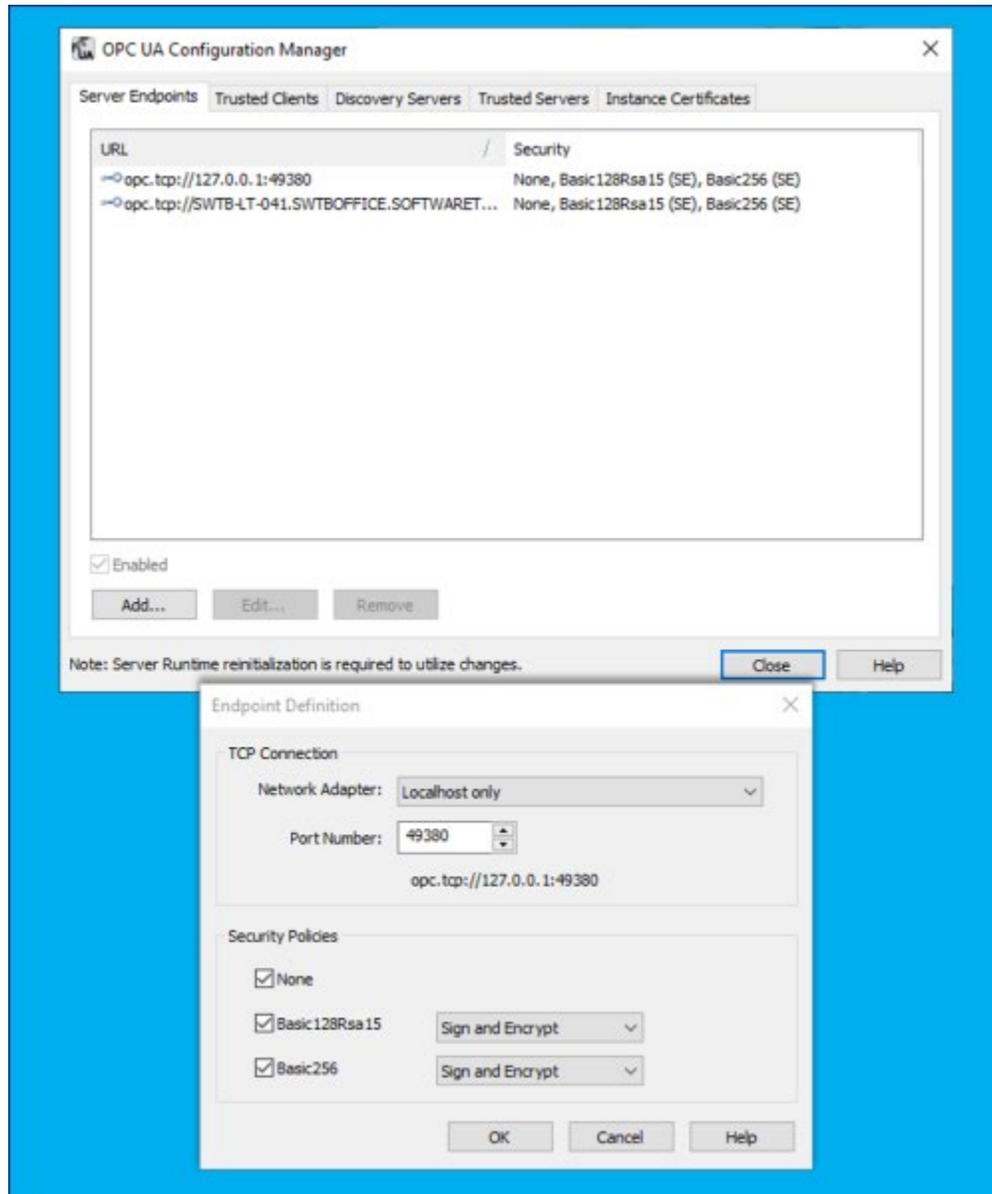
#### 1. **Server Endpoints**

For an OPC UA server, the endpoint is how an OPC UA client specifies a connection. For those familiar with OPC DA, this would be equivalent to the OPC DA Server ProgID at a very basic level. This section is where you configure the server endpoints that you would

like to be available to your OPC UA client applications.

A server endpoint consists of the syntax “opc.tcp://” followed by either the IP address or Hostname of the machine where the OPC UA server is installed, followed by a colon and then the Port Number, which is configurable in the endpoint in TOP Server.

2. You can specify the Network Adapter the endpoint should use (which will determine the IP address or Hostname that is used) or if the connection should only allow local OPC UA clients to connect to it.



Additionally, the endpoint is where you define the level of secure encryption options that

an OPC UA client applications must support and use to make a connection to that endpoint. These options are updated as technology advances – currently, options include Basic256Sha256, Basic256, Basic128Rsa15, and None (ranging from highest level of security to no security at all and not recommended).

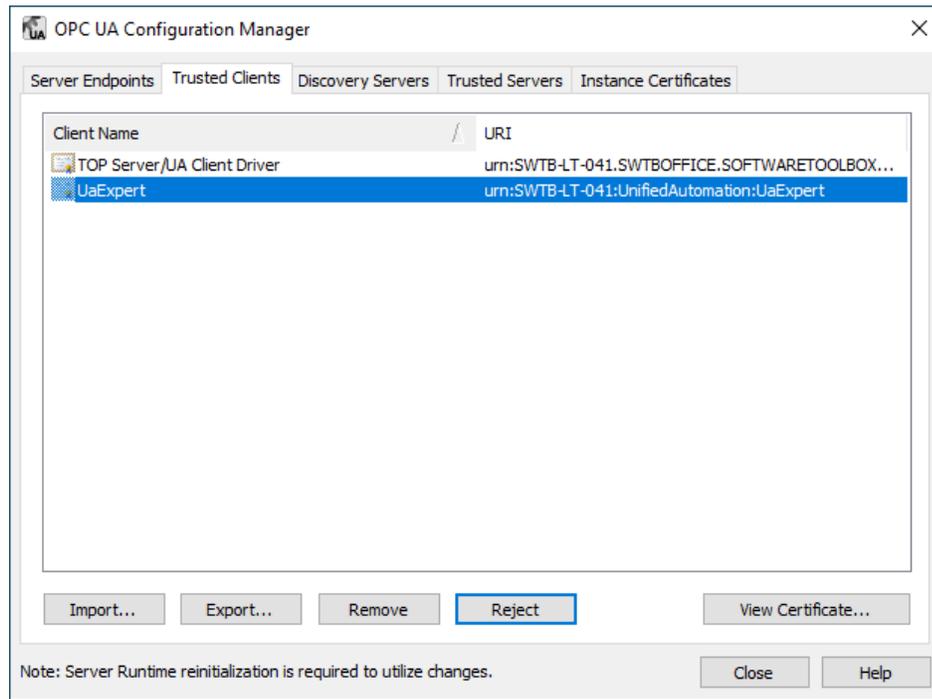
For each option (other than None, of course), you can also select whether the endpoint requires Sign and/or Sign and Encrypt (for full details on Sign and Encrypt, refer back to the [Then What is Sign & Encrypt section](#) of this e-book).

Up to 100 separate endpoints can be configured in TOP Server, so you can provide specific endpoints for specific UA clients with the appropriate level of security and encryption for your application.

It is recommended to only define as many endpoints as you need and to disable any endpoints that you have defined but are not currently using. Disabling an endpoint is as simple as highlighting the desired endpoint and unchecking the “Enabled” setting at the bottom of the tab.

### 3. **Trusted Clients**

In order for an OPC UA client to connect to TOP Server, it must be trusted – this means that the UA client and TOP Server have exchanged security certificates and you, the user, have told TOP Server it is okay for that UA client to connect to it. That is accomplished here, in the Trusted Client section of the OPC UA Configuration Manager.



Here you can manually Import the certificate from your OPC UA client (consult your client’s help documentation on how to manage its certificates including where they are stored and how to export them). You can also Export certificates listed here or Remove them completely.

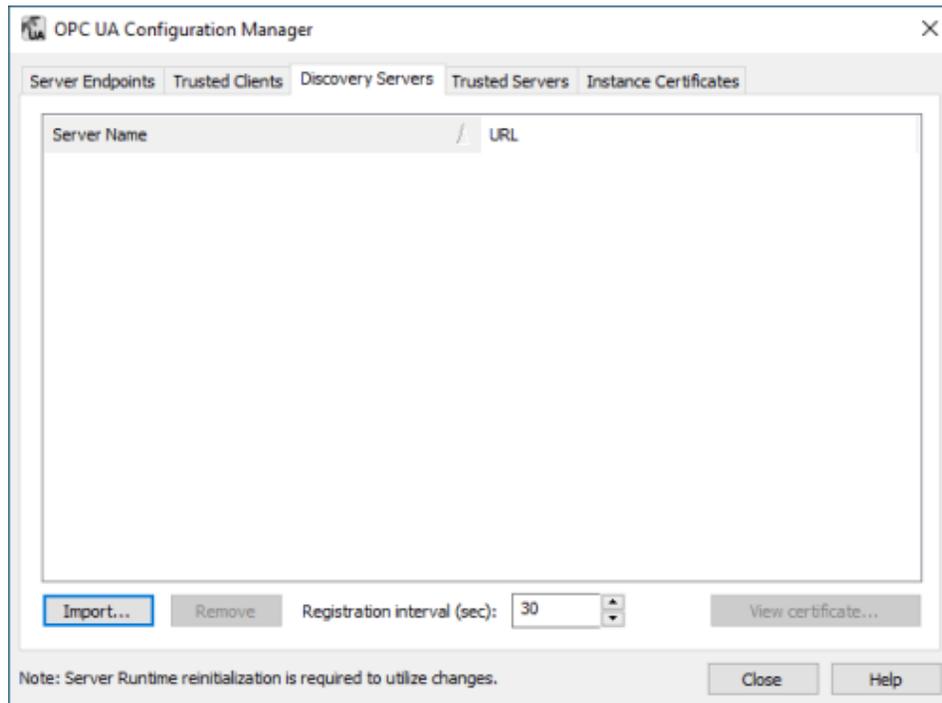
And to the point of trusting a client, here you can choose to Trust a client’s certificate for clients that have attempted to connect without first exchanging certificates or, if you change your mind about that trust later, you can Reject any clients that show up here.

You can even View the certificate information of these UA clients in this section.

#### 4. **Discovery Servers**

And, while TOP Server doesn’t install with one, the Discovery Servers section is available to define any Local Discovery Server (LDS) services that you may have installed either on the same machine as TOP Server or on another machine that is network accessible by TOP Server. This includes the Local Discovery Server available from the [OPC Foundation \(available to registered users\)](#).

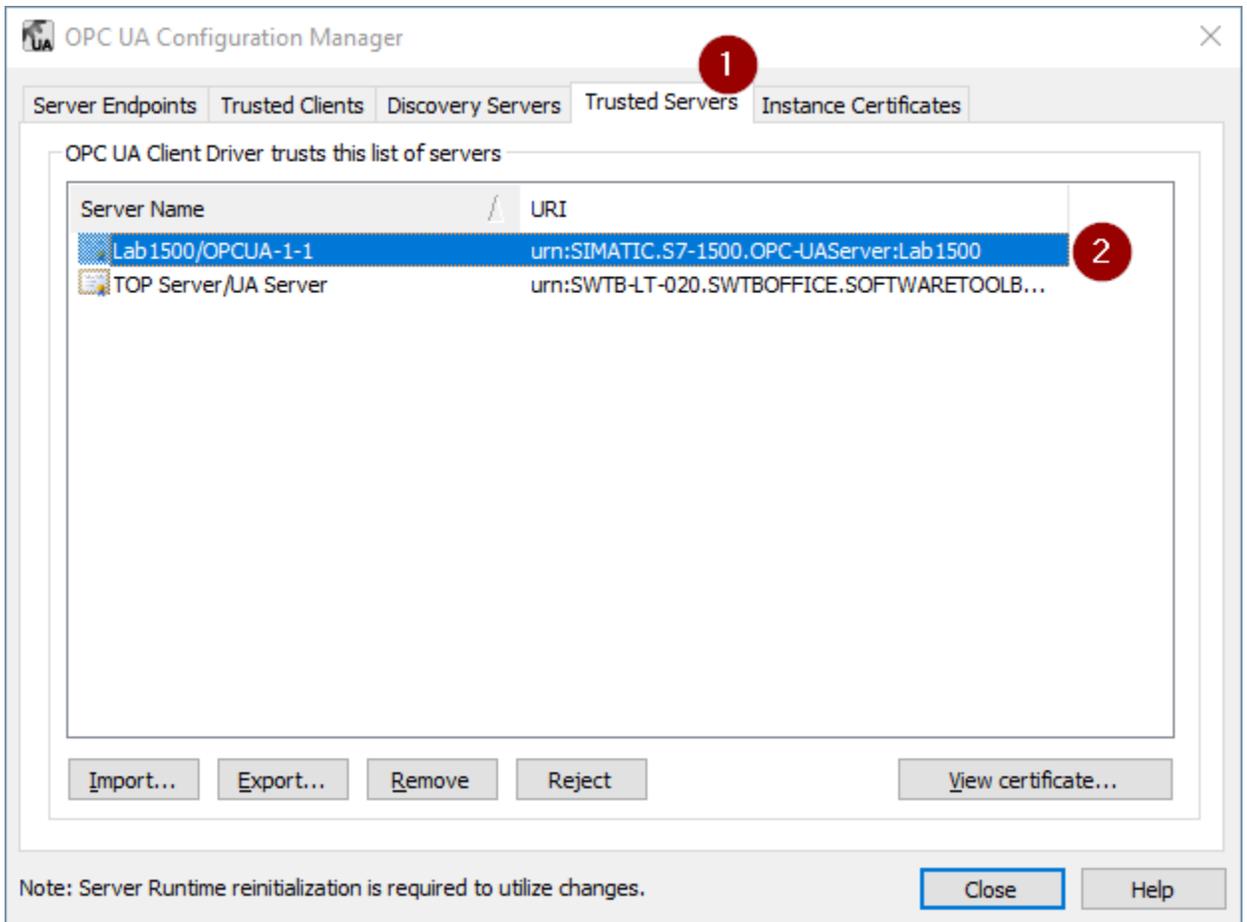
A discovery server for OPC UA is a dedicated service with which an OPC UA server can register, allowing OPC UA clients to then look at the LDS to “browse” for available OPC UA servers they can connect to. Again, for those familiar with OPC DA Classic, this is similar to what you might be used to with OPCEnum for browsing OPC DA servers.



Basically, you will need the security certificate from the LDS (which will include the UA endpoint for the LDS) and you'll import that in this section. You can easily Remove an LDS later or View its certificate information. And the frequency that TOP Server will re-register with the LDS is configurable (it defaults to every 30 seconds).

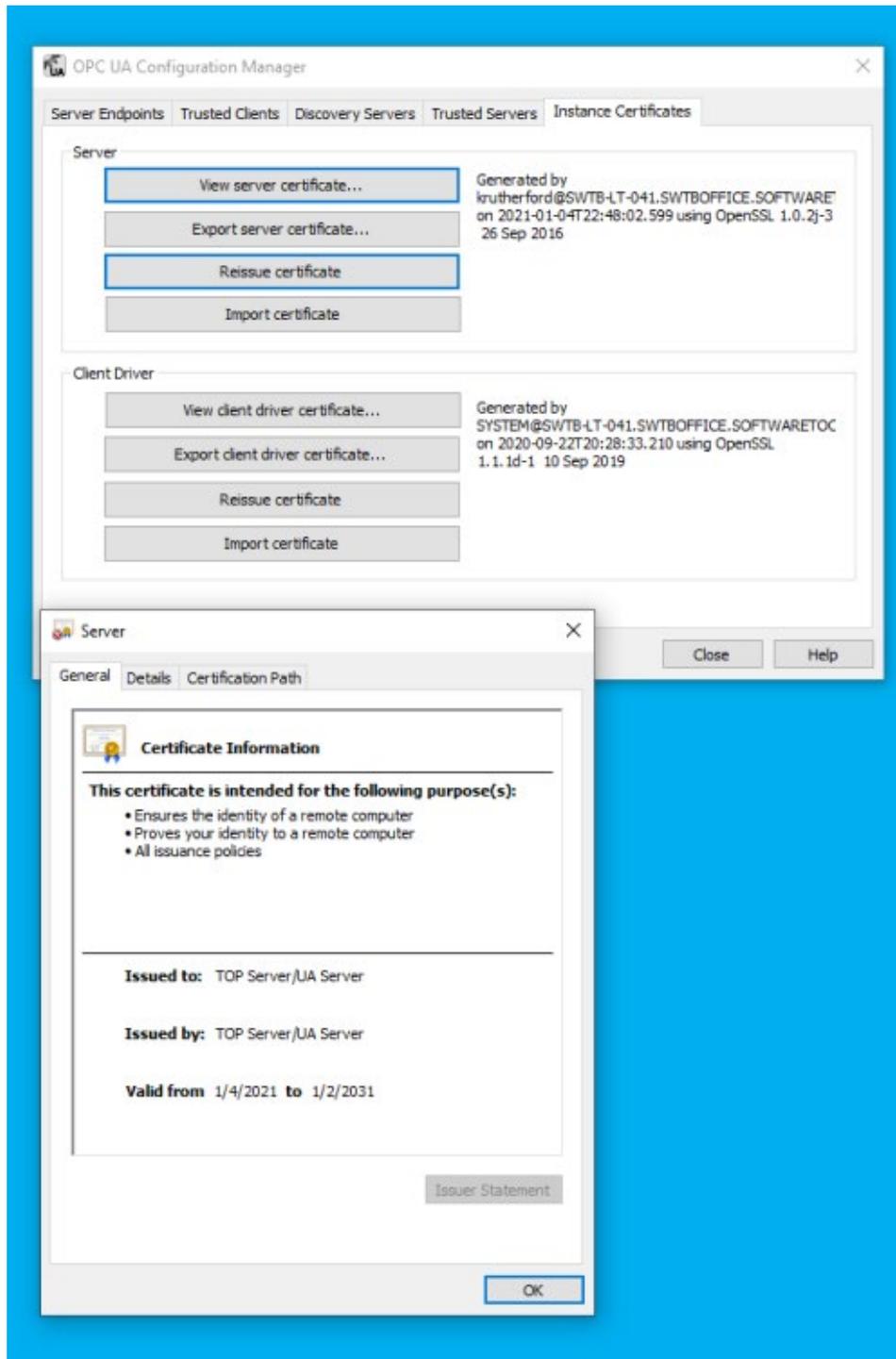
## 5. **Trusted Servers**

As we just discussed with the trust relationship required for an OPC UA client to connect to TOP Server, that same relationship is necessary for TOP Server to connect to another OPC UA server using the [OPC UA Client driver](#). This section is where those certificates are managed in similar fashion except that, here, you're managing and trusting or rejecting certificates from other OPC UA servers instead of OPC UA clients.



## 6. Instance Certificates

And last but not least, this is where you manage the OPC UA client and server certificates for TOP Server. By default, when you install TOP Server, a self-signed certificate is generated for the OPC UA server interface (and the OPC UA Client driver, if installed).



This section lets you manage those certificates, providing the ability to View the information for both certificates, Export both certificates, Import both certificates and, most importantly, Reissue both certificates.

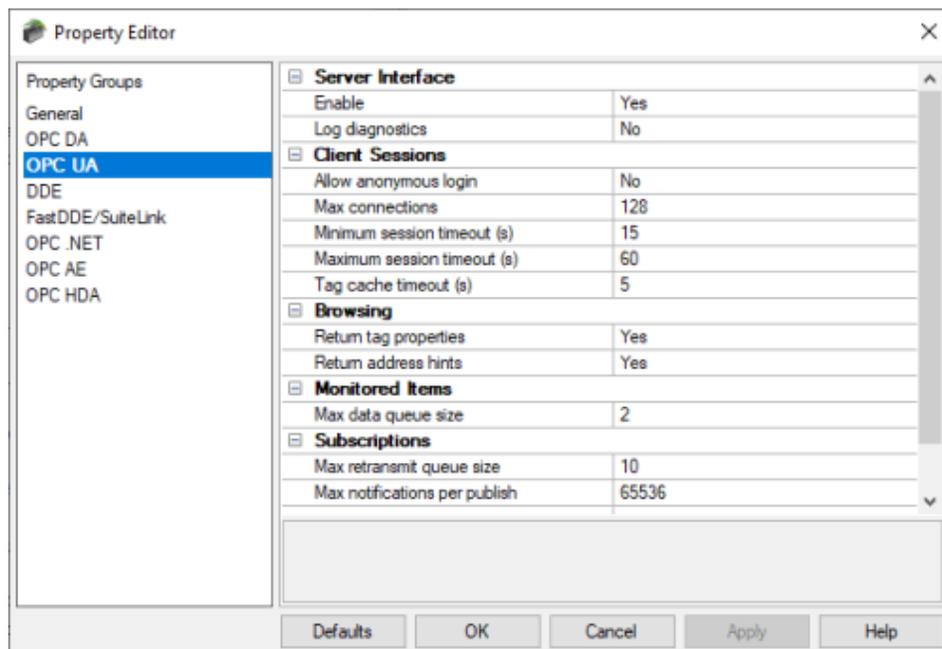
Reissuing a certificate is most important because certificates eventually expire and have to be reissued (this operation also then requires a re-exchanging and trust of those certificates with OPC UA clients and servers that you've previously made that exchange with. Starting with TOP Server V6.7, self-signed certificates expire after 3 years from the issue date. As such, be aware that if you're using OPC UA, you'll need to reissue and exchange your certificates every 3 years.

Import provides you with the ability to utilize a third-party certificate from a certificate authority such as Verisign, Thawte, etc. in the event that you prefer not to use the self-signed certificates issued by TOP Server.

Export provides you with the ability to manually export your certificates for manual exchange with OPC UA clients and servers to establish the trust relationship we've been discussing.

## 2. TOP Server Project Properties

In the TOP Server Configuration, there are also settings related to the OPC UA server interface located in the Project Properties (right-click on "Project" in the tree view and select "Properties"). In the OPC UA section, the majority of these settings can be left at the default values but I want to focus on the 3 most important ones that you'll need to be aware of here.



Key properties to be aware of:

1. **Enable**

By default, for the highest level of security out of the box, the TOP Server OPC UA server interface is disabled. As such, in order to actually connect an OPC UA client to TOP Server, you'll need to change toggle this setting to "Yes".

2. **Log diagnostics**

TOP Server supports a powerful diagnostics tool for viewing and capture OPC transactions – this can be used for OPC UA, OPC DA and OPC AE client connections (accessed under View -> OPC Diagnostics). By default, though, diagnostics for each of those interfaces is disabled. To capture OPC UA diagnostics using this feature, you need to toggle this setting to "Yes".

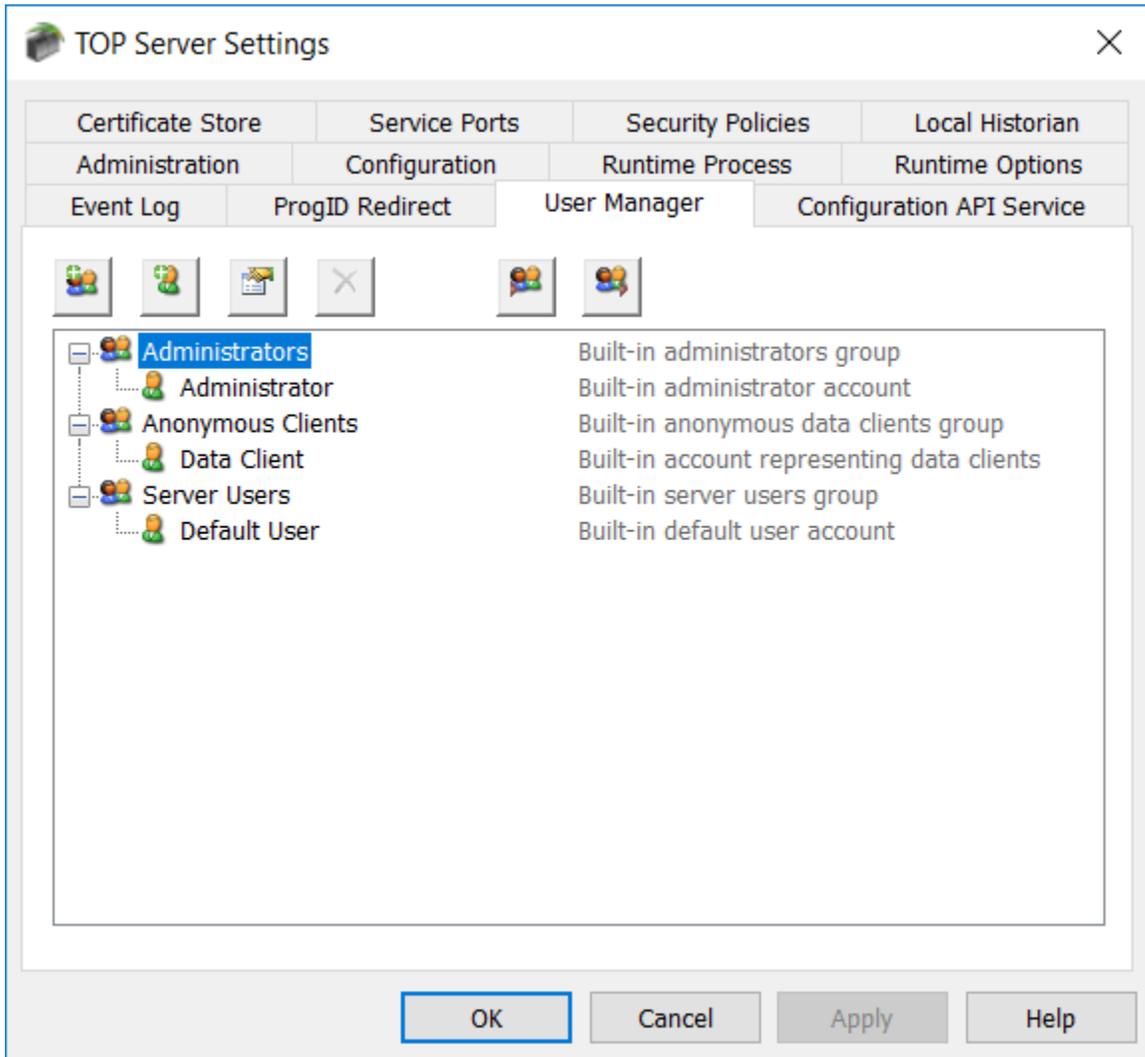
3. **Allow anonymous login**

By default, TOP Server requires that an OPC UA client authenticate by providing a username and password (defined in the TOP Server User Manager, which we'll be covering next) that is authorized to access TOP Server. While not recommended for live production operations, it is supported to allow UA clients to connect to TOP Server anonymously by toggling this setting to "Yes". So if your TOP Server is on a machine that is not accessible from the internet and your IT team is comfortable that your network is secure enough for this, you can allow anonymous access.

### ***3. TOP Server User Manager***

As we just mentioned briefly, TOP Server has a built-in User Manager that is accessible from the TOP Server Administration system tray icon (right-click and select "Settings" then "User Manager"). The User Manager provides a number of benefits including auditability in your TOP Server event log and granular assignment of permissions by user (even down to the tag level when using the Security Policies plug-in).

For the context of this discussion, users of OPC UA clients will need at least one user defined with permissions to access TOP Server via OPC UA.



Permissions are defined at the group level in the User Manager, so that means the user can belong to any defined group (Administrators, Server Users, User Defined) as long as that group allows the user the required access to the tags you need to read and/or write in TOP Server. Options include:

- I/O Tag Access (including the ability to use dynamic addressing where static tags are not define in your devices in TOP server)
- System Tag Access (accessing the special properties and statistics tags available at the server, channel and device levels)
- Internal Tag Access (access special internal tags available for certain drivers)

- Browse Project Namespace (for browsing and selecting static tags, including system, statistics and internal tags).

For more details on using the User Manager and the Security Policies plug-in, we recommend [reviewing our blog post on the subject here](#).

## Connecting Your OPC UA Client to TOP Server

Using the information we've just discussed, you can get your OPC UA client connected to TOP Server. You'll want to step through the following list, as a general rule (or [watch our tutorial video on connecting an OPC UA Client to TOP Server here](#)):

1. Enable the following in the TOP Server Project Properties:
  - a. OPC UA server interface
  - b. OPC UA Diagnostics
  - c. Anonymous login (if you prefer not to use user authentication – otherwise, leave this disabled)
2. Configure an endpoint in the OPC UA Configuration Manager:
  - a. Specify the desired port number.
  - b. Specify the desired encryption settings (keep in mind what your OPC UA client supports – not all OPC UA clients/servers will always support the same levels of encryption so it's important to select an options that both the client and server support).
3. Export the security instance certificate from your OPC UA client (how to do this varies by client application – please consult the help documentation for your OPC UA client for details on exporting its OPC UA security instance certificate).
4. Import the security instance certificate from your OPC UA client in the TOP Server OPC UA Configuration Manager under “Trusted Clients” by clicking the “Import” button and browsing to the certificate file from Step 4 above.
5. Export TOP Server's security instance certificate in the TOP Server OPC UA Configuration Manager under “Instance Certificates” by clicking the “Export server certificate” button.
6. Import TOP Server's security instance certificate into your OPC UA client (how to do this varies by client application – please consult the help documentation for your OPC UA client for details on importing server security instance certificates and trusting them).

7. Configure a user with the required permissions in the TOP Server User Manager, ensuring you have Read permissions allowed for I/O Tags at a minimum and Dynamic Addressing if you don't have static tags defined in your TOP Server project.
8. Configure a connection from your OPC UA client using the endpoint address:
  - a. You'll need the URL from the endpoint that you configured in TOP Server (as displayed under the Port setting in the Endpoint – you can copy/paste from there).
  - b. You'll need to know which security policy you enabled in the endpoint so you can select the appropriate option in your client.
  - c. You'll need the username and password that you configured in the TOP Server User Manager (if applicable – if you enabled Anonymous login, you can leave the username and password blank in your UA client).
  - d. Consult your client's help documentation on the specific steps required to complete the connection and how to either browse for static nodes/tags in TOP Server (if you have them configured) or how to manually add nodes/tags.

For a walkthrough of connecting an OPC UA client application to TOP Server, [watch the detailed how-to video here](#).

### Connecting TOP Server to another OPC UA Server

Again using the information we discussed earlier, you can get your OPC UA client connected to TOP Server. You'll want to step through the following list, as a general rule (or [watch our tutorial video on connecting an TOP Server to other OPC UA servers here](#)):

1. Make sure your other OPC UA server is properly configured to accept OPC UA client connections in a similar fashion to the steps suggested above for the TOP Server OPC UA server interface including enabling the interface (if applicable) having your OPC UA endpoint configured and any username/password authentication setup properly (consult the help documentation for your other OPC UA server for details on preparing for OPC UA clients to connect including how to export the security instance certificate).
2. You'll need the following details from your other OPC UA server in order to configure the TOP Server OPC UA Client driver to connect:
  - a. OPC UA endpoint URL (including Port)
  - b. Security policies that are supported and enabled (including whether Sign and/or Sign & Encrypt are required)
  - c. The security instance certificate from that OPC UA server.

3. Import the security instance certificate from your other OPC UA server in the TOP Server OPC UA Configuration Manager under “Trusted Servers” by clicking the “Import” button and browsing to the certificate file from Step 1 above.
4. Export the security instance certificate for the OPC UA Client driver in the TOP Server OPC UA Configuration Manager under “Instance Certificates” by clicking the “Export client driver certificate” button.
5. Import the TOP Server OPC UA Client driver security instance certificate into your OPC UA server and trust it ([consult the help documentation for your other OPC UA server to determine how to import and trust client certificates](#)).
6. Configure a new channel in your TOP Server Configuration (which represents the OPC UA server you’re connecting to):
  - a. Select the “OPC UA Client” from the dropdown for the type of channel.  
(**NOTE:** If that option is not available, you’ll need to modify your TOP Server installation and add the OPC UA Client driver – [click here for a how-to video](#)).
  - b. Give the channel meaningful name – this will be part of any item references of upstream clients connecting to this TOP Server to access data from your other OPC UA server.
  - c. Keep the defaults in the next section.
  - d. Enter the Endpoint URL from your other OPC UA server (including port)
  - e. Select the Security Policy to be used for the connection.
  - f. Select whether the connection will use Sign or Sign and Encrypt for the connection.
  - g. Keep the defaults in the next section.
  - h. If required, enter the appropriate Username and Password for your OPC UA server endpoint in the other OPC UA server (leave blank if connecting anonymously).
  - i. Review the changes you’ve made in the summary and click Finish if everything looks correct.
7. Add a new device to the channel you just created in your TOP Server Configuration:
  - a. Give the device a meaningful name – this will also be part of any item references of upstream clients connecting to this TOP Server to access data from your other OPC UA server.

- b. Keep the defaults for the next 6 sections until you get to the “Select Import Items” portion of the configuration.
    - If your underlying OPC UA server has items/nodes configured and is browsable, you can click the “Select import items” button to browse your server’s address space and select items/nodes to import for access by your upstream client applications.
    - If the server does not have items/nodes available or doesn’t support browsing, you can click Next to skip this step and manually add items/nodes after completing the device configuration.
  - c. Review the changes you’ve made in the summary and click Finish if everything looks correct.
8. Now that the channel and device are complete, if you weren’t able to browse for items/nodes in your underlying OPC UA server but you need to add static tags, you can do so now by right-clicking on the device you just configured and selecting “New Tag”.
- a. The Tag Name should be a meaningful name that you want to associate with the item/node from your other OPC UA server.
  - b. For the address, the syntax here will largely depend on your underlying OPC UA server – consult the documentation for that server to determine the correct addressing required from an OPC UA client application.
  - c. In general, the TOP Server UA Client driver uses addressing syntax composed of a Namespace Index, followed by a Type and ending with a Value that depends on the selected Type.
    - For instance, a valid address syntax for accessing an item/node in TOP Server via OPC UA would be  
ns=3;s=ChannelName.DeviceName.TagName
    - For more details on possible address syntaxes in the UA Client driver, consult the [OPC UA Client driver help file](#) under “Address Descriptions”.
    - Your underlying OPC UA server documentation, as I mentioned, should provide an indication of which namespace and type you should use along with the required values.
    - It is also possible to skip defining static tags in TOP Server completely and simply define dynamic references in your upstream client application – you will still need to determine the above information about your OPC UA server to know how to configure your dynamic addresses.

Once your channel, device and items/nodes are complete, you can quickly test your configuration using the OPC Quick Client that installs with TOP Server by launching it from the TOP Server Configuration window toolbar.

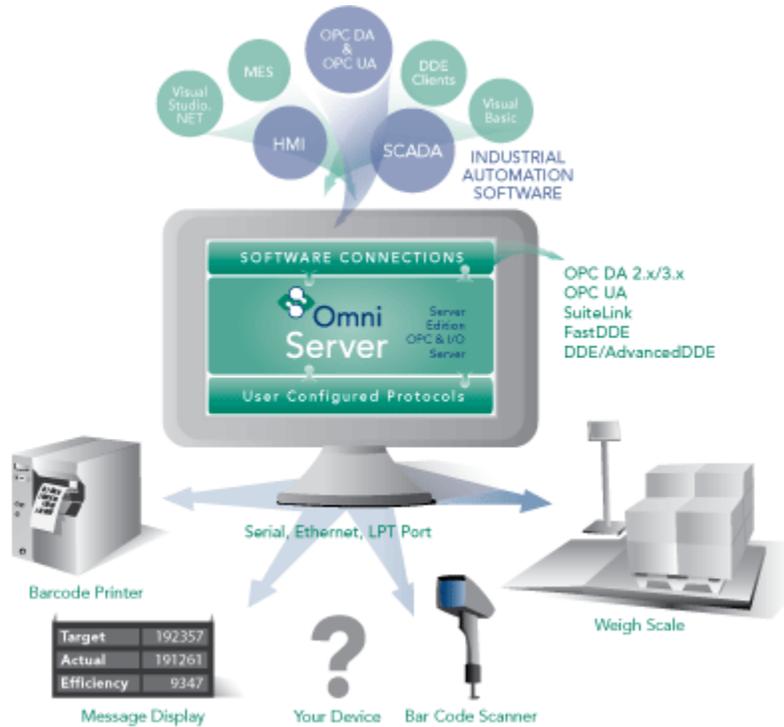
For a walkthrough of connecting TOP Server to another OPC UA server, [watch the detailed how-to video here](#). Additionally, [our article on additional use cases for the OPC UA Client driver](#) has other useful information. As always, you can try out everything we've just covered yourself with the [free trial of TOP Server](#).

For security purposes in general, to ensure your TOP Server implementation is secure as possible, we also recommend reviewing the TOP Server Secure Deployment Considerations Guide. This guide covers a range of security topics including network environment, host operating system, installation, post-installation and other security considerations when installing and configuring TOP Server. You can [download the TOP Server Secure Deployment Considerations Guide here](#).

And another important resource is the [TOP Server Video Resources web page](#). It contains a number of detailed how-to videos on a range of topics related to configuring TOP Server.

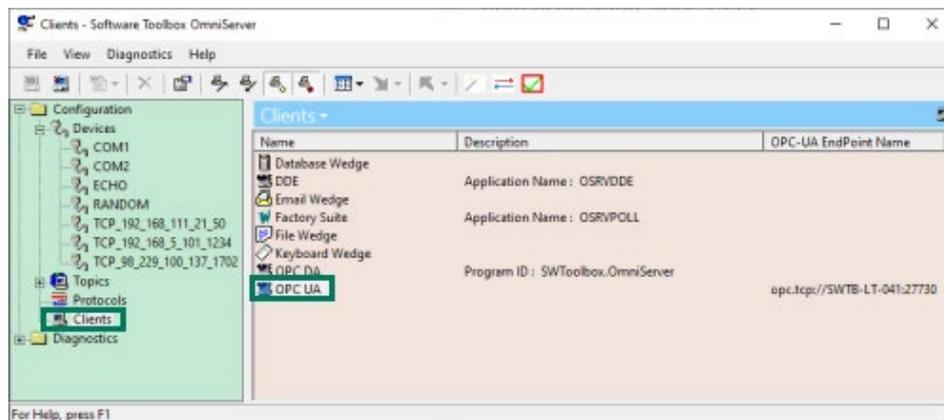
## OmniServer OPC UA Security (Server)

With [OmniServer](#), both the [Server](#) and [Professional Editions](#) natively provide an OPC UA server interface for access by OPC UA client applications such as HMI, SCADA, MES, Historians and other applications that need to access process data from non-standard device such as scales, barcode scanners, RFID readers, sensors and more. That OPC UA server interface is available at no additional cost with those editions of OmniServer.

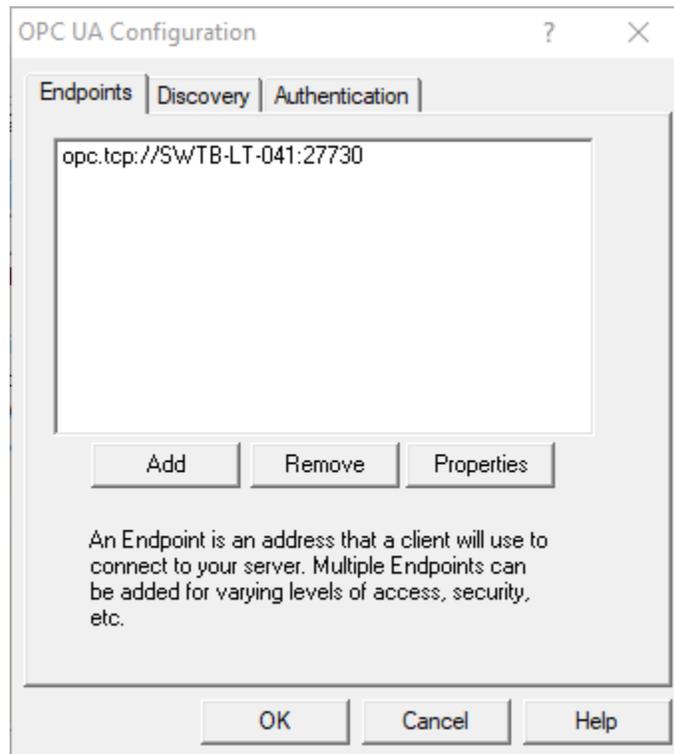


## OPC UA Configuration Components in OmniServer

The settings relevant to OmniServer's OPC UA server interface are all located in the OPC UA section of the **Clients** view in the OmniServer Configuration window.



Simply double-click on "OPC UA" to launch the "OPC UA Configuration" which has the following settings with specific purposes for UA connections:



### ***1. OmniServer OPC UA Endpoints***

This section is where you configure the OmniServer OPC UA server endpoints that you would like to be available to your OPC UA client applications.

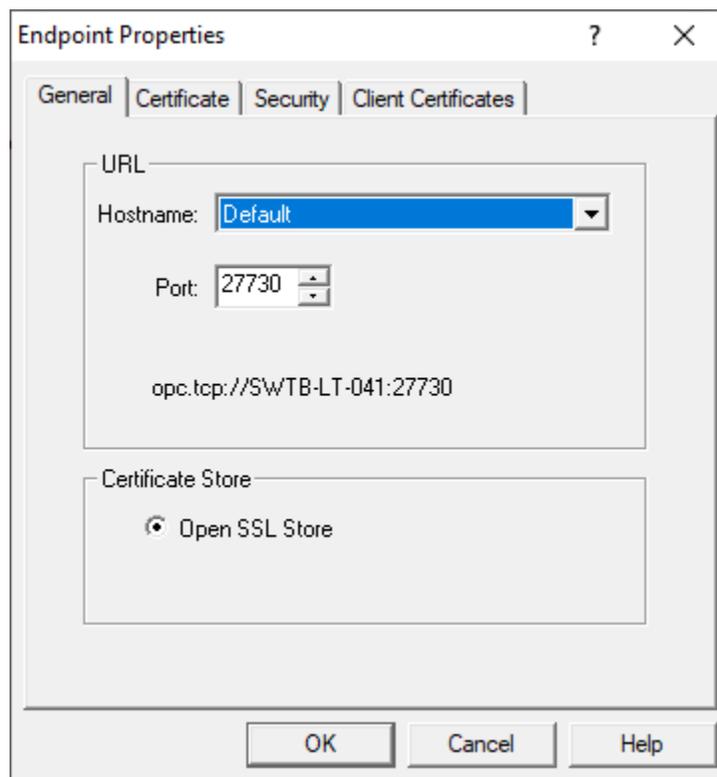
In OmniServer, there is an existing default endpoint with a default port of 27730 accessible both remotely and locally for OPC UA clients. You can select that existing endpoint and click "Properties" to make edits or click "Add" to add a new endpoint with different settings (you can add multiple endpoints with their own specific levels of security and access to meet the needs of your project).

You can also "Remove" any existing endpoint that you no longer need (it is recommended to only have configured the endpoints you need and are using).

The **OmniServer Endpoint** properties consist of the following:

1. **General Settings**

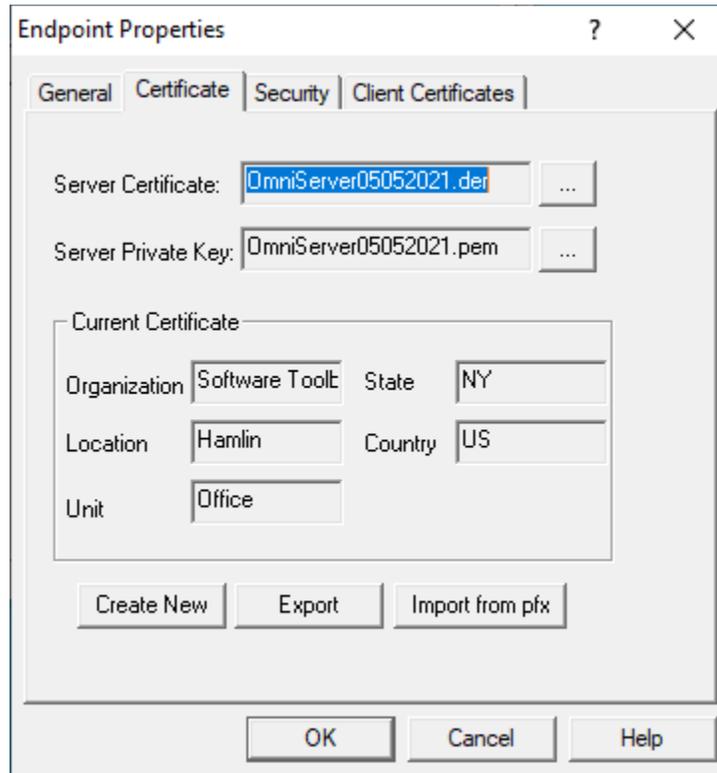
A server endpoint consists of the URL, which is composed of the starting syntax “opc.tcp://” followed by either the IP address or Hostname of the machine where the OPC UA server is installed, followed by a colon and the Port Number, which is configurable in the endpoint in OmniServer. You can also specify "Local Only" for the "Hostname" field if the connection should only allow local OPC UA clients to connect to it (this results in the URL using the loopback IP address of 127.0.0.1 with the configured port number).



OmniServer uses the OpenSSL store for storing the OPC UA certificates exchanged with OPC UA clients for secure communications.

## 2. Certificate Settings

This is where you manage OmniServer's own OPC server certificate. By default, when you install OmniServer, a self-signed certificate is generated for the OPC UA server interface.



This section lets you manage that certificate, providing the ability to **Create New** certificates (which is how you re-issue the existing self-signed certificate), **Export** the certificate (useful for manual import into your OPC UA client) or **Import from pfx** (which gives you the ability to use a security certificate that you're purchased from a third-party certificate authority such as Verisign, Thawte, etc in the event that you prefer not to use the self-signed certificate)..

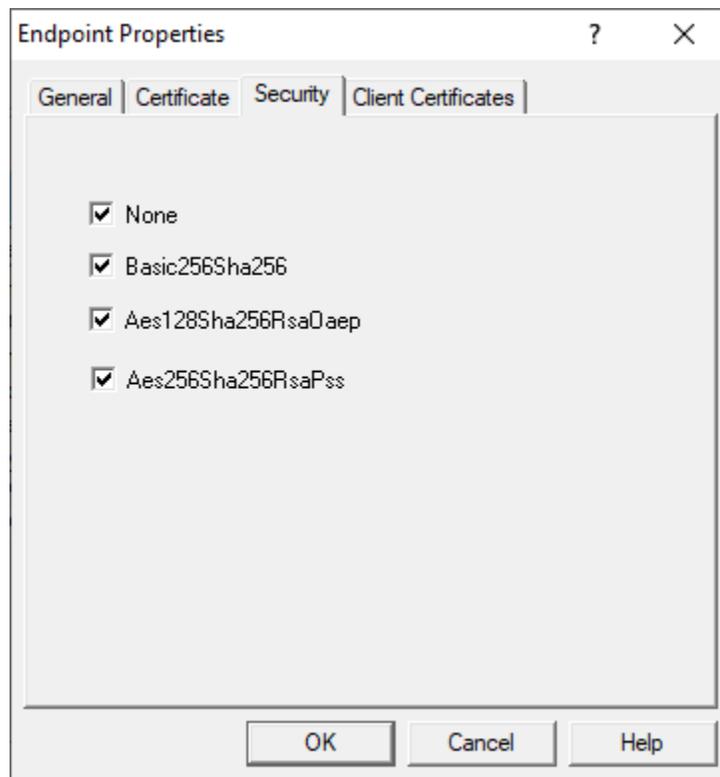
Reissuing a certificate (using the "Create New" button) is most important because certificates eventually expire and have to be reissued (this operation also then requires a re-exchanging and trust of those certificates with OPC UA clients that you've previously made that exchange with. OmniServer self-signed certificates expire after 3 years from the issue date. As such, be aware that if you're using OPC UA, you'll need to reissue and exchange your certificates every 3 years.

Use the "Export" button here to export OmniServer's current certificate file if you prefer to import the certificate to your OPC UA client and trust it first prior to attempting a

connection to OmniServer (how to do this varies by client application – please consult the help documentation for your OPC UA client for details on importing server security instance certificates and trusting them).

### 3. **Security Settings**

This section is where you define the level of secure encryption options that an OPC UA client applications must support and use to make a connection to this endpoint. These options are updated as technology advances – currently, options include: **Aes256Sha256RsaPss**, **Aes128Sha256RsaOaep**, **Basic256Sha256**, and **None** (ranging from highest level of security to no security at all and not recommended).

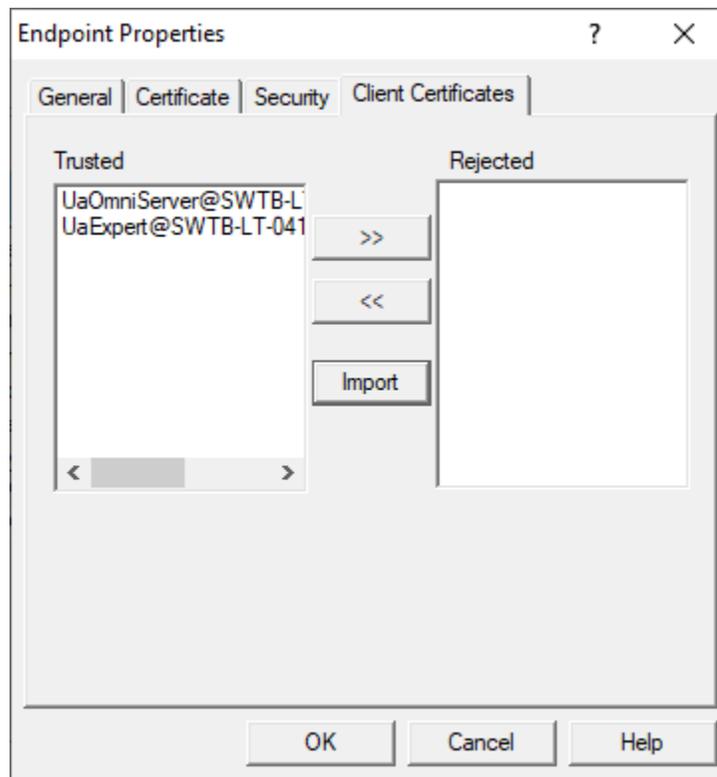


For each option (other than None, of course), both Sign and/or Sign and Encrypt and encrypt are supported (for full details on Sign and Encrypt, refer back to the [Then What is Sign & Encrypt section](#) of this e-book). By default, all security options are enabled for an endpoint - after testing, it is typically recommended to disable "None" for production systems for the highest level of security. It is recommended to consult your IT department for network and security best practices specific to your systems.

#### 4. Client Certificates Settings

For an OPC UA client to connect to OmniServer, it must first be trusted – this means that the UA client and OmniServer have exchanged security certificates and you, the user, have told OmniServer it is okay for that UA client to connect to it (and, vice versa, you've told your UA client that it's okay to connect to OmniServer).

The first part, telling OmniServer your OPC UA client is okay, is accomplished in this section.



Here you can manually **Import** the certificate from your OPC UA client (consult your client's help documentation on how to manage its certificates including where they are stored and how to export them).

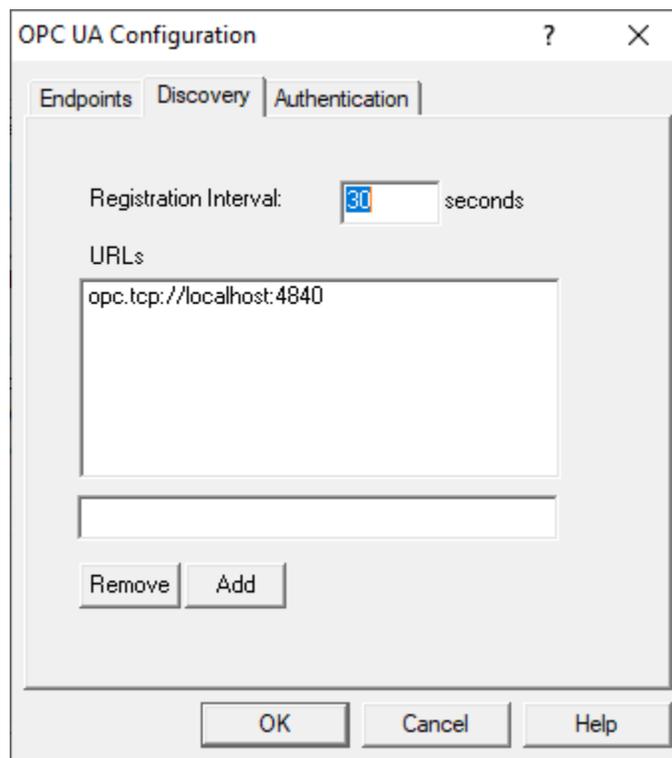
The Trusted section here is where OPC UA clients will appear if you have trusted their certificates. If an OPC UA client attempts to connect and has not yet been trusted, it will appear in the Rejected section.

You can use the ">>" arrow button to untrust a previously trusted UA client and the "<<" arrow button to trust a previously untrusted UA client that has attempted to connect to OmniServer.

## 2. OmniServer OPC UA Discovery Settings

While OmniServer (like most OPC UA solutions) doesn't install with one, the Discovery section of the OPC UA Configuration is available to define any Local Discovery Server (LDS) services that you may have installed either on the same machine as OmniServer or on another machine that is network accessible by OmniServer. This includes the Local Discovery Server available from the [OPC Foundation \(available to registered users\)](#).

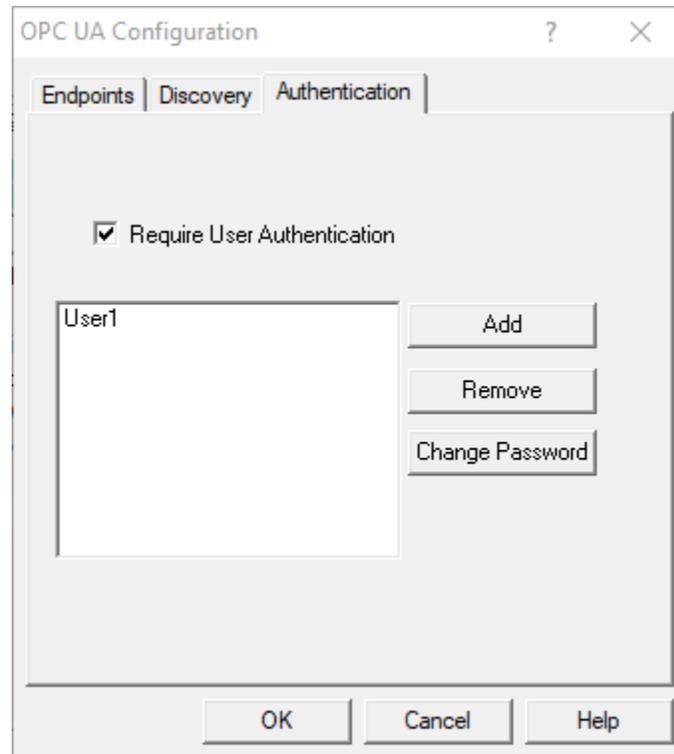
A discovery server for OPC UA is a dedicated service with which an OPC UA server can register, allowing OPC UA clients to then look at the LDS to "browse" for available OPC UA servers they can connect to. Again, for those familiar with OPC DA Classic, this is similar to what you might be used to with OPCEnum for browsing available OPC DA servers.



Basically, you will need the endpoint URL for the LDS and you'll enter that URL in the field under the URLs list and click the **Add** button. The configurable "Registration Interval" defaults to 30 seconds and defines how frequently OmniServer will attempt to register with the URLs you've added below. You can easily **Remove** an LDS later by highlighting it in the URLs list and clicking "Remove".

### 3. OmniServer OPC UA Authentication (Log-in) Settings

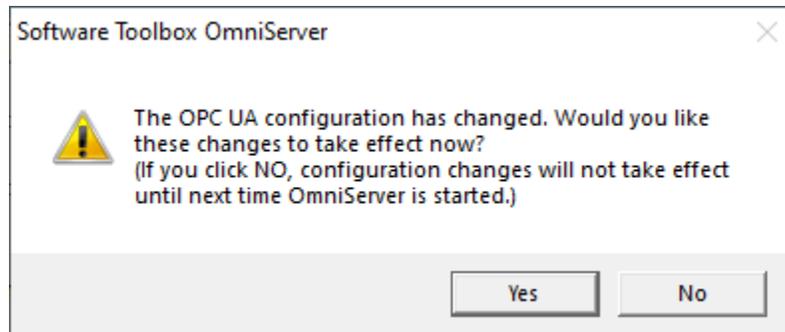
By default, OmniServer doesn't require an OPC UA client to provide a username and password when connecting.



In the "Authentication" section, you can enable this requirement by checking the "Require User Authentication" setting. Once enabled, you'll be able to use the provided buttons to:

- **Add** - Add a username and associated password that a user can specify from a trusted OPC UA client to connect to OmniServer.
- **Remove** - Deletes the currently highlighted user from the list of existing users.
- **Change Password** - Allows you to change the password of the currently highlighted user from the list of existing users.

After making any and all changes to any of the sections in the OmniServer OPC UA Configuration, click the "OK" button to apply the settings. You will receive the following prompt - clicking "Yes" will apply your OPC UA setting changes immediately. If you click "No", your settings won't apply until the next time the OmniServer runtime is restarted.

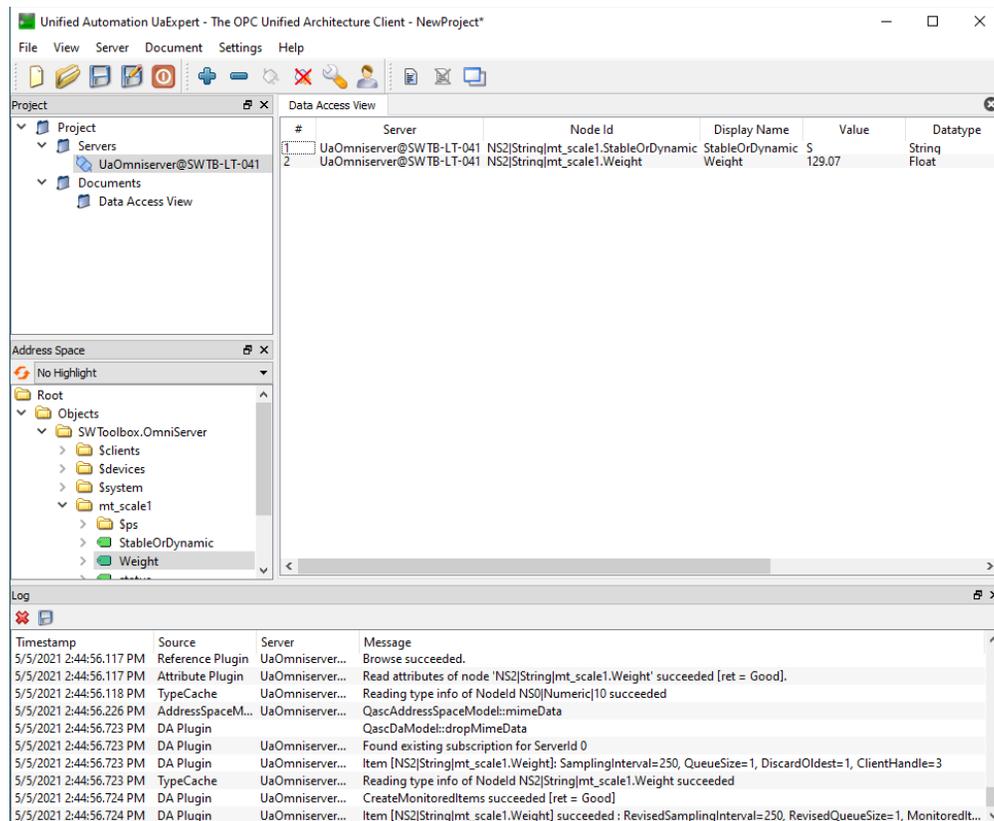


## Connecting Your OPC UA Client to OmniServer

Using the information we've just discussed, you can get your OPC UA client connected to OmniServer. You'll want to step through the following list, as a general rule (or [watch our tutorial video on connecting an OPC UA Client to OmniServer here](#)):

1. Configure an endpoint in the OmniServer OPC UA Configuration:
  - a. Specify the desired port number.
  - b. Specify the desired encryption settings (keep in mind what your OPC UA client supports – not all OPC UA clients/servers will always support the same levels of encryption so it's important to select an options that both the client and server support).
2. Export the security instance certificate from your OPC UA client (how to do this varies by client application – please consult the help documentation for your OPC UA client for details on exporting its OPC UA security instance certificate).
3. Import the security instance certificate from your OPC UA client in the OmniServer OPC UA Configuration in the Endpoint properties under "Client Certificates" by clicking the "Import" button and browsing to the certificate file from Step 2 above.
4. Export OmniServer's security instance certificate in the OmniServer OPC UA Configuration under "Certificate" in the endpoint you plan to connect to by clicking the "Export" button.
5. Import OmniServer's security instance certificate into your OPC UA client (how to do this varies by client application – please consult the help documentation for your OPC UA client for details on importing server security instance certificates and trusting them).

6. Configure a user and password in the OmniServer OPC UA Configuration under "Authentication".
7. Configure a connection from your OPC UA client using the endpoint address:
  - a. You'll need the URL from the endpoint that you configured in OmniServer (as displayed in the General section of the Endpoint Properties).
  - b. You'll need to know which security policies you enabled in the OmniServer endpoint you'll be connecting to so you can select the appropriate option in your OPC UA client.
  - c. You'll need the username and password that you configured in the Authentication section of the OmniServer OPC UA Configuration (if applicable – if you didn't enable "Require User Authentication" in OmniServer, you will leave the username and password blank in your UA client).
  - d. Consult your client's help documentation on the specific steps required to complete the connection and how to either browse for static nodes/tags in OmniServer (if you have them configured) or how to manually add nodes/tags.

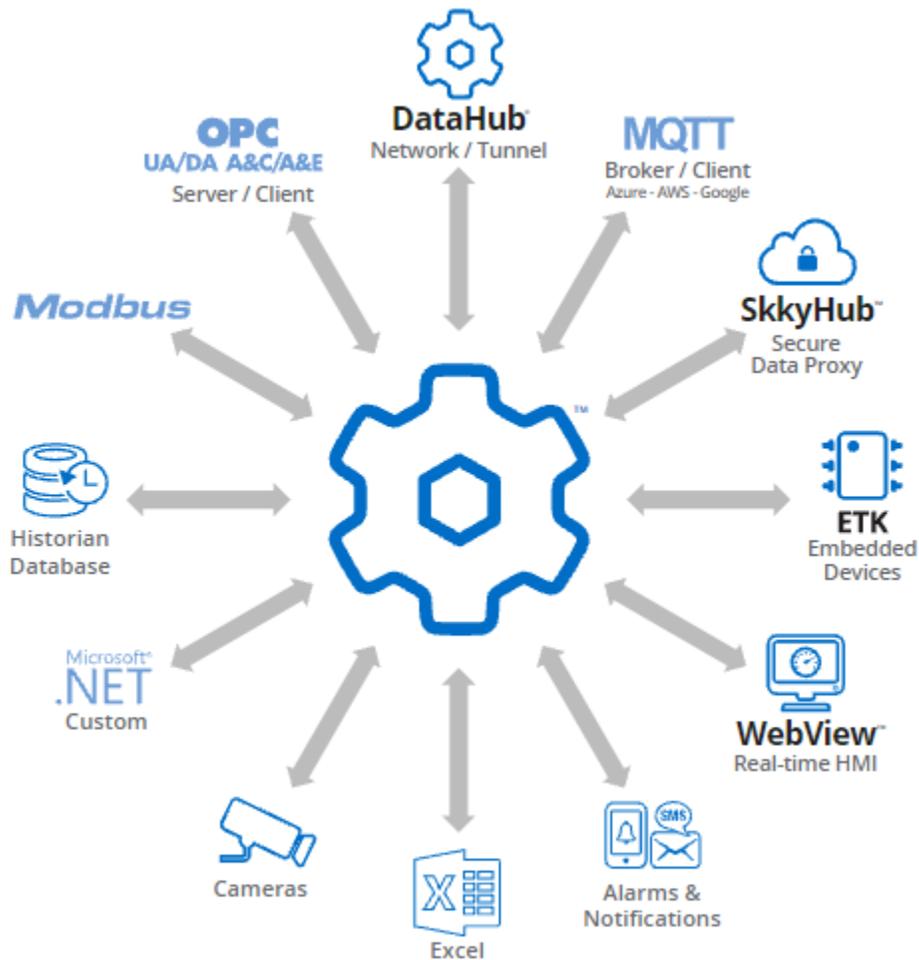


For a walkthrough of connecting an OPC UA client application to OmniServer, [watch the detailed how-to video here](#). And another important resource is the [OmniServer Video Resources](#)

[web page](#). It contains a number of detailed how-to videos on a range of topics related to configuring OmniServer.

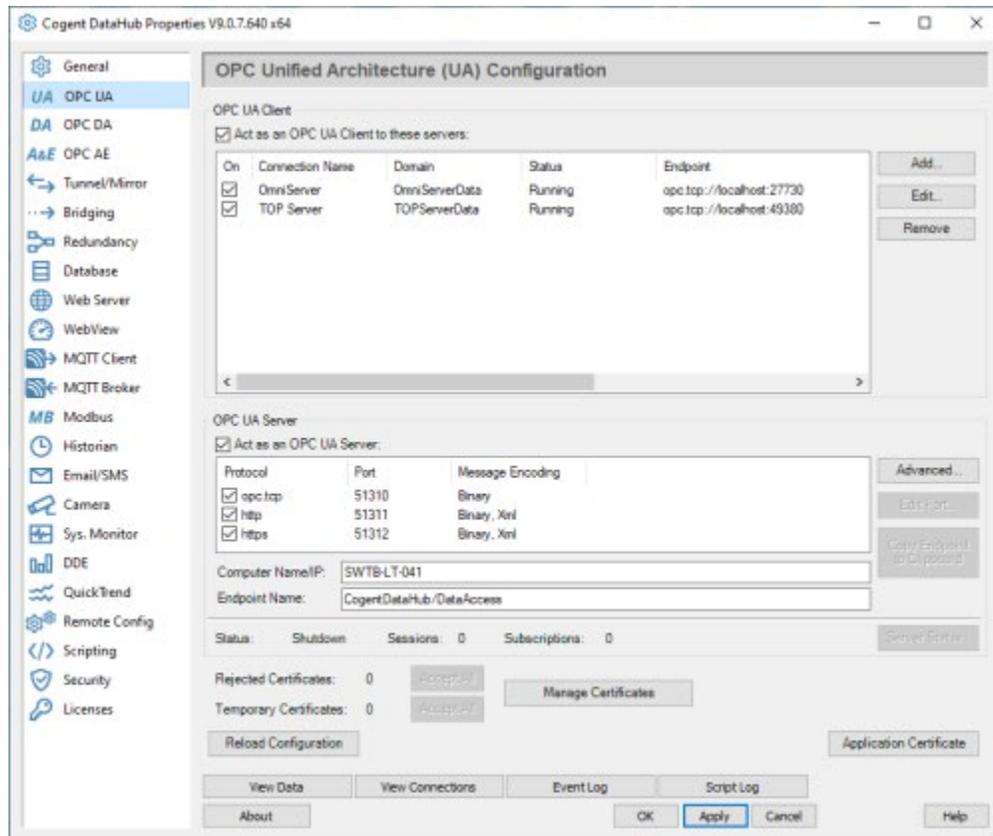
## Cogent DataHub OPC UA Security (Client & Server)

[Cogent DataHub](#), by nature of being a [gateway between different interfaces](#), can act as both an [OPC UA client and an OPC UA server](#). This, coupled with those other interfaces, allows conversion to and from OPC UA for a variety of different systems that don't natively support OPC UA yet.



## OPC UA Configuration Components in Cogent DataHub

The settings relevant to OPC UA (both client and server) in the DataHub are all located in the OPC UA section of the configuration.



### 1. DataHub OPC UA Client Settings

In the consolidated OPC UA settings in DataHub shown above, the top portion of the section covers adding, editing and removing connections from DataHub acting as an OPC UA client to one or more OPC UA servers.

Adding or editing a OPC UA server that DataHub will be connecting to results in the following properties dialog:

Clicking "Add" or "Edit" brings up the following available settings for connecting DataHub to an OPC UA server:

1. **Connection Name** - User-defined friendly name used by DataHub to identify this connection.
2. **Discovery Domain** - While DataHub doesn't install with one, this setting is available to define any Local Discovery Server (LDS) services that you may have installed either on the same machine or on another machine that is network accessible by DataHub. This includes the Local Discovery Server available from the OPC Foundation ([available to registered users](#)).

A discovery server for OPC UA is a dedicated service with which an OPC UA server can register, allowing OPC UA clients to then look at the LDS to “browse” for available OPC UA servers they can connect to. For those familiar with OPC DA Classic, this is similar to what you might be used to with OPCEnum for browsing OPC DA servers.

If you have an available LDS, here is where you would enter the IP or computer name for where the LDS is located - this allows the "Endpoint URL" setting to be populated with OPC UA servers that have registered with that LDS for selection.

If an LDS is not available, you can just enter the IP or computer name where the OPC UA server is located.

3. **Endpoint URL** - For an OPC UA server, the endpoint is how an OPC UA client specifies a connection. For those familiar with OPC DA, this would be equivalent to the OPC DA Server ProgID at a very basic level. This setting is the URL for the OPC UA server you want to connect to - you can either select it from the dropdown list, if available for browsing (per the Discovery Domain setting) or you can manually enter the correct URL.

A server endpoint consists of the syntax “opc.tcp://” followed by either the IP address or Hostname of the machine where the OPC UA server is installed, followed by a colon and then the Port Number, which may or may not be configurable in the endpoint of the OPC UA server you're connecting to. You can specify the Network Adapter the endpoint should use (which will determine the IP address or Hostname that is used) or if the connection should only allow local OPC UA clients to connect to it.

Additionally, the endpoint is where the level of secure encryption options that the OPC UA client applications must support and use to make a connection to that endpoint. You'll need to confirm what sign and encrypt options are supported for your OPC UA server.

Generally, for each supported encryption option (other than None, of course), you can also select whether the endpoint requires Sign and/or Sign and Encrypt (for full details on Sign and Encrypt, refer back to the [Then What is Sign & Encrypt section](#) of this e-book).

4. **Security Policy** - For connecting to other OPC UA server, DataHub currently supports (in order of most to least secure): Basic256Sha256, Basic 256, Basic128Rsa15, or None (Default).

6. **User Token Type** - This setting specifies what user authentication method, if any, should be used to connect to the OPC UA server. DataHub supports the following options:
  - a. My Certificate - this option uses the self-signed security certificate generated by DataHub for use with authentication - When selected, the certificate name will be displayed in the "My Certificate" field.
  - b. Another Certificate - this option allows you to import a different security certificate file (such as from a 3rd party certificate authority such as Verisign or Thawte) for use with authentication - When selected, the "Certificate File" field will allow you to browse to the .pfx certificate file.
  - c. User Name - this option requires a user name and password (corresponding to a user configured in the OPC UA server) - when selected, you can enter the User Name and Password in the resulting fields.
  - d. Anonymous (Default) - this option, which must also be supported by the UA server, doesn't use any authentication for connecting to the OPC UA server.
7. **Always accept invalid server certificate** - Disabled by default, when enabled this setting tells DataHub to basically ignore whether the OPC UA server's security certificate is invalid now or in the future. (NOTE: This setting is provided as a flexible convenience and should only be enabled with the understanding that it lowers the security of your OPC UA connection).
8. **Continue to accept the server certificate when it expires** - Enabled by default, this option allows a UA certificate to be accepted outside of its valid time window, meaning that expired certificates can continue to be used. Keeping this setting enabled also keeps the OPC UA server and client connected if their respective system clocks ever get out of synch. Additionally, you'll want to keep this setting enabled if you're connecting to an OPC UA server running on an embedded system without a system clock, or where the system clock cannot be kept in sync.
9. **Do not disconnect when the server reports a failed state** - Disabled by default, so if the OPC UA server is in a non-RUNNING state, DataHub disconnects and logs a message to the Event Log. Enabling this setting overrides that behavior and maintain the connection to the OPC UA server.

11. **Connection Test** - Once you've configured the Endpoint URL, Security Policy and User Token Type, you can click this button to initiate a test connection to the OPC UA server to confirm those settings are correct (if there are any issues, a meaningful message indicating what the problem was will appear in the Message section of the dialog).
12. **Maximum Update Rate** - This setting specifies the fastest data will be requested from the underlying OPC UA server and is useful for slowing down the rate of incoming data. The minimum value is 10 and the default is 1000 ms. This value is also used as the polling for non-subscription options selected in the "Read Method" setting.
13. **Read Method** - This setting defines how DataHub will read data from the underlying OPC UA server. The following options are supported (in order of most efficient to least efficient):
  - a. Subscription (Default) - This is the recommended option - updates are requested from the OPC UA server that are event-based (i.e. updates should only be sent if the value for a point actually changes).
  - b. Asynchronous Read - A non-blocking read method where DataHub will send a read request at the "Maximum Update Rate" but doesn't stop performing other operations while waiting for the response.
  - c. Synchronous Cache Read - A blocking read method where DataHub will send request for the latest cached value for points from the last time a device read was performed and wait for the response.
  - d. Synchronous Device Read - A blocking read method where DataHub will send a read request at the "Maximum Update Rate" for the latest value from the device and waits for the response before performing other operations.
14. **Write Method** - This setting defines how DataHub will write data to the underlying OPC UA server. The following options are supported (in order of most efficient to least efficient):
  - a. Asynchronous Write - DataHub performs a write/writes to points in the OPC UA server and does not wait for a response.
  - b. Synchronous Write (Default) - DataHub performs a write/writes to points in the OPC UA server and waits for a response prior to performing other operations - this can be useful with UA servers that either don't support async writes or that can't handle a large quantity of async writes. This can also be useful if order of operations with your writes matters (such as with writing recipe information).
15. **Monitored Item Queue Size** - The maximum number of items between polls that get stored for this server.
16. **Maximum Request Item Count** - The OPC UA spec allows a UA server to specify the number of items it will allow per request. This setting allows you to adjust the DataHub to what the server allows, if necessary. Defaults to 500 items.

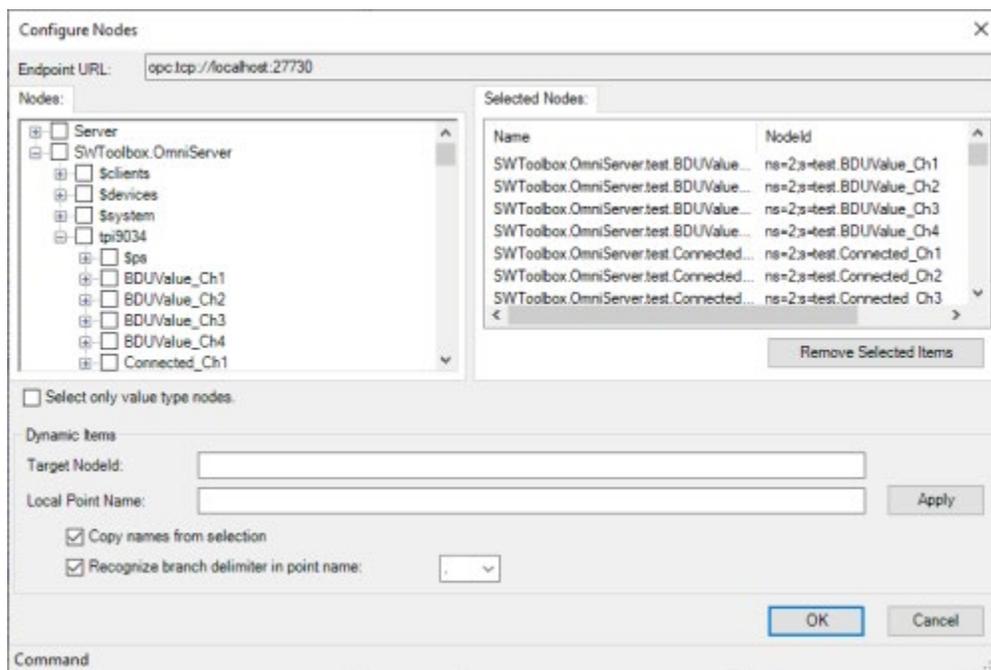
17. **Only transmit GOOD quality data to this server** - Setting restricts point updates from the DataHub to the server to only those with "Good" quality.
18. **Do not accept null data from this server** - Setting rejects point updates from the server to the DataHub if the values are null.
19. **Create multiple subscriptions using Maximum Request Item Count** - The "Maximum Request Item Count" specifies the maximum number of nodes (points) per subscription. Coupled with this setting being enabled (default), DataHub will use that number to decide the maximum number of nodes per subscription.

However, if this number is small and the total number of nodes is large then the number of requested subscriptions could exceed the subscription count limit of the server.

Unchecking this box will solve that problem by putting all of the nodes into a single subscription.

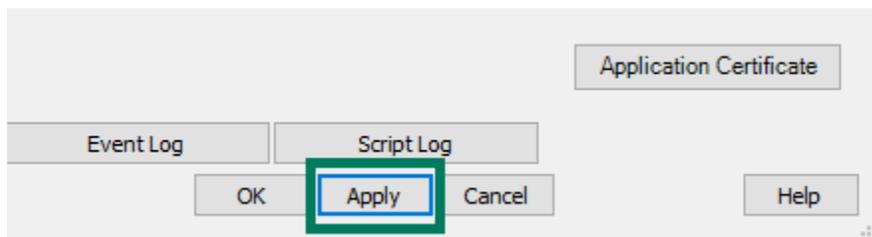
20. **Advanced** - The settings available in the Advanced section covered a number of timeout and length limits - it is recommended not to adjust these timing settings, as the defaults work best in the majority of situations. As such, they are beyond the scope of this post.
21. **Load All Nodes on Server** - Enabling this option will result in all available nodes (points/items) available in the underlying OPC UA server being loaded in DataHub (not recommended unless you need to access every available node/point).
22. **Manually Select Nodes** - Enabled by default, this allows you to click "Configure Nodes" and manually browse and select the nodes/points you wish to access with DataHub.

23. **Configure Nodes** - Click this button to open the "Configure Nodes" dialog where you can browse and select one or more nodes/points from the OPC UA server to access.



24. **Data Domain Name** - This is the DataHub domain or group where the nodes/points from this OPC UA server will be placed and available for access by other functional modules and interfaces in DataHub.

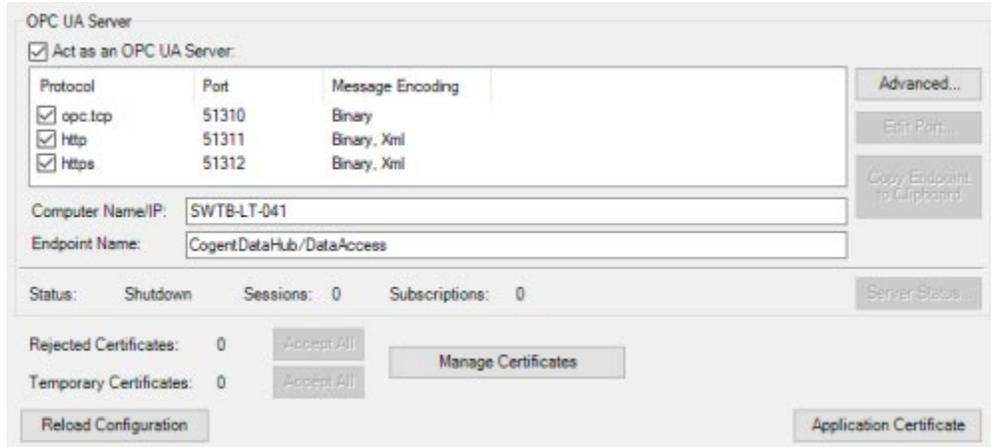
As with other sections and settings in the DataHub, once you've edited the OPC UA settings for connecting to an OPC UA server, always make sure to "OK" to save and exit the dialog. Then make sure, once back in the main DataHub configuration interface, to click the "Apply" button to ensure the settings you changed take effect.



This includes making sure to enable the checkbox "Act as an OPC UA Client to these servers" and clicking "Apply" - otherwise, DataHub will not connect to your configured OPC UA server/servers. Additionally, if you ever need to disable just one of multiple connections to OPC UA servers, you can uncheck the box next to each configured connection and click "Apply".

## 2. DataHub OPC UA Server Settings

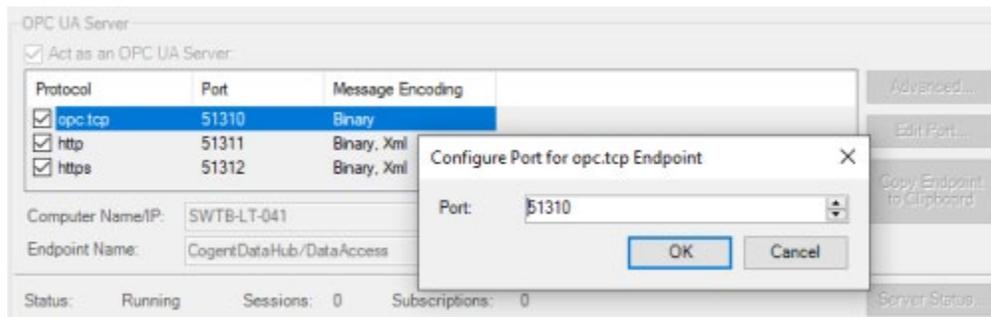
In the consolidated OPC UA settings in DataHub, the bottom portion of the section covers the settings related to OPC UA client applications connecting to DataHub.



These settings determine what gets specified in your OPC UA client: Clicking "Add" or "Edit" brings up the following available settings for connecting DataHub to an OPC UA server:

1. **Act as an OPC UA Server** - When unchecked, DataHub's OPC UA server interface will not be available for OPC UA clients to connect to - it should be checked if you plan to connect other OPC UA clients to DataHub.
2. **Protocols** - By default, all supported OPC UA protocols are enabled in DataHub - the most common is opc.tcp, but DataHub also supports connections from OPC UA clients via http and https transports, as well.

Each protocol has a different TCP port, which is configurable by highlighting the desired protocol and clicking the "Edit Port" button.



3. **Computer Name/IP** - Automatically populates with the local Computer Name for use in the endpoint that OPC UA clients will use to access DataHub.
4. **Endpoint Name** - An optional meaningful name that can be appended to the OPC UA server endpoint that OPC UA clients will use to access DataHub. This defaults to

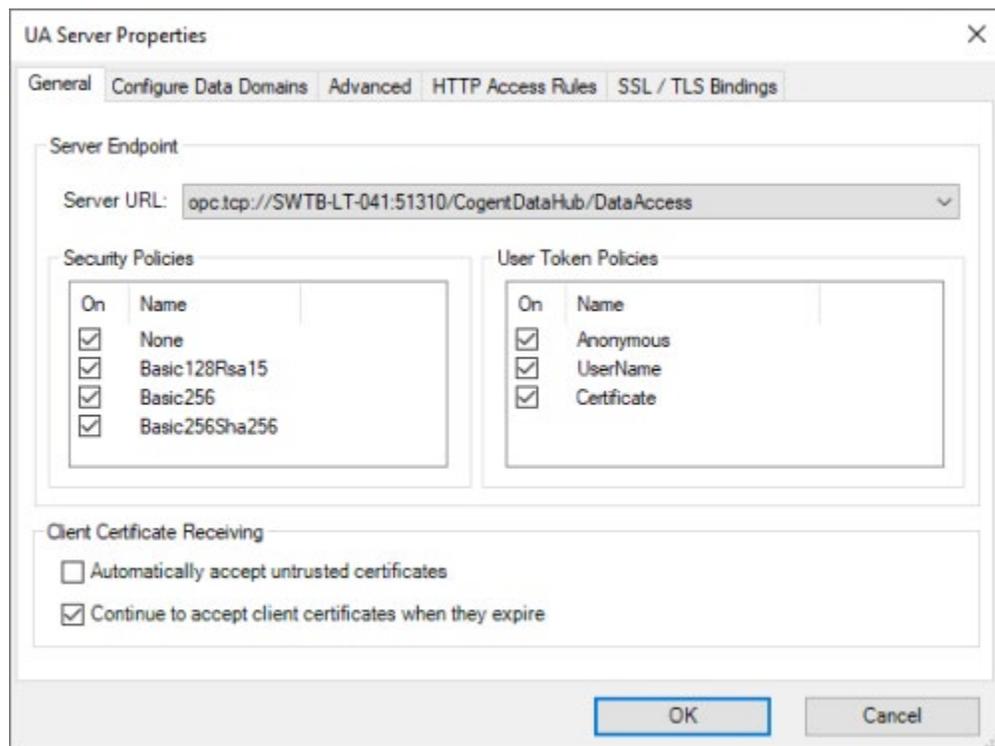
"CogentDataHub/DataAccess" but can be edited to the desired label.

5. **Copy Endpoint to Clipboard** - When you highlight the desired "Protocol" option, you can click this button to copy the fully qualified OPC UA endpoint to ensure you have the right syntax. This makes it easy to transfer the right endpoint string to your OPC UA client. For example:

*opc.tcp://SWTB-LT-041:51310/CogentDataHub/DataAccess*

6. **Advanced** - Most of the sections in the Advanced options are beyond the scope of this e-book and should be left at their defaults. We will cover the "General" section, which is where the security policies and authentication settings are defined for each endpoint and several relevant settings in the "Advanced" section.

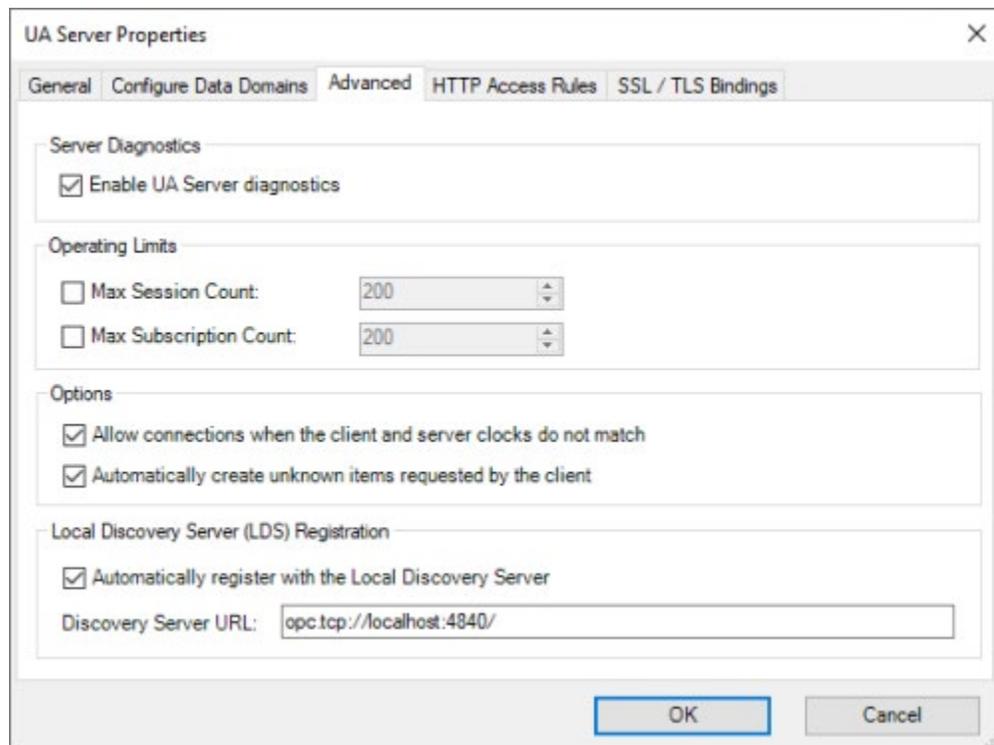
- a. **General**



- i. Server URL - Dropdown for selecting the endpoint you wish to configure.
- ii. Security Policies - Select the security policies that will be available for OPC UA clients to use when connecting to DataHub - by default, all supported options are enabled including (in order from most to least secure): Basic256Sha256, Basic256, Basic128Rsa15, None.

- iii. User Token Policies - Select the user authentication options that will be available for OPC UA clients to use when connecting to DataHub - by default, all supported options are enabled including Certificate, UserName and Anonymous.
- iv. Automatically accept untrusted certificates - Disabled by default, this setting allows DataHub to automatically trust certificates from OPC UA clients attempting to connect to DataHub (for greatest security, it is recommended to leave this disabled - this option is provided as a convenience for testing purposes).
- v. Continue to accept client certificates when they expire - Enabled by default, this setting allows DataHub to maintain connections with OPC UA clients even after the client's security certificate has expired (this is not recommended for production systems, for security purposes, and is enabled only for ease of initial testing).

**b. Advanced**

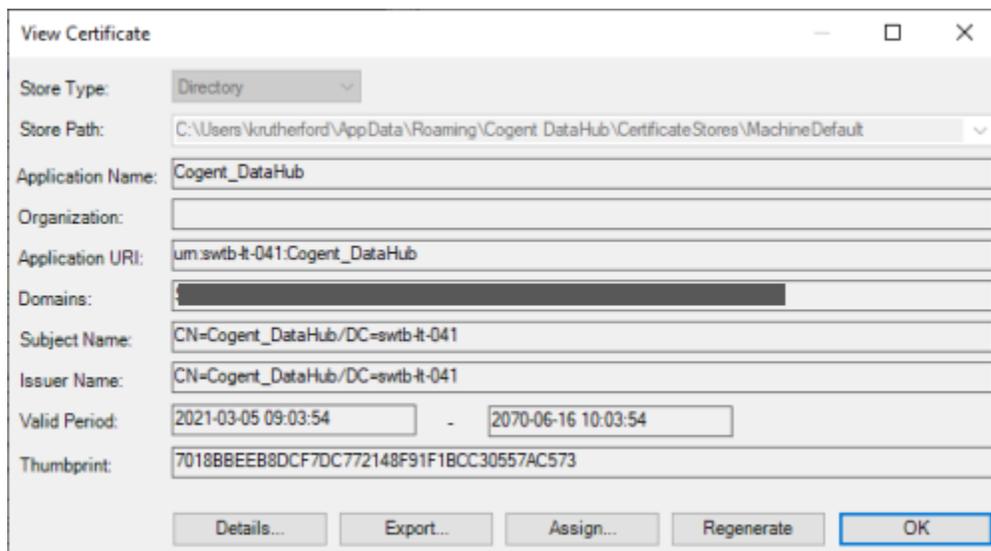


- i. Enable UA Server diagnostics - Enabled by default, this allows OPC UA server specific messages, warnings or errors to be logged to the main DataHub Event Log for easier troubleshooting - it is recommended to

keep this enabled.

- ii. Allow connections when the client and server clocks do not match - Enabled by default, this setting ensures that there are no issues with OPC UA clients connecting from machines where there may be a difference in system clock time (this eliminates the need to synchronize the system clocks of the two systems).
  - iii. Automatically create unknown items requested by client - Enabled by default, this allows DataHub to dynamically add any node/point requested by an OPC UA client that doesn't already exist in DataHub - for full OPC compliance, this setting should be disabled, which will result in an error being returned to any OPC UA client requesting nodes that don't already exist.
  - iv. Automatically register with the Local Discovery Server - If a Local Discovery Server is available, enabling this option, when combined with a valid "Discovery Server URL" for that LDS will result in DataHub registering with that LDS and being browsable by OPC UA clients accessing that LDS.
  - v. Discovery Server URL - The OPC UA endpoint URL of an available Local Discovery Server service (either on the same machine as DataHub or on a different machine) where DataHub can register to be browsable by OPC UA client applications.
7. **Manage Certificates** - Clicking this button brings up the dialog for managing security certificates in the various available certificates stores for DataHub's OPC UA server interface. Selecting the certificate store from the dropdown will display any certificates in that store.
- a. Rejected Certificates - where any OPC UA client certificates will be listed that have attempted to connect but have not been trusted or that you have explicitly rejected by using the "Reject" button in any other certificate store (or certificates that you have manually imported using the Import button in the "Rejected Certificates" store.
  - b. OPC UA Private Certificates - where OPC UA certificates that have been trusted in the private (non-Windows) certificate store will be listed.
  - c. OPC UA Global Certificates - where OPC UA certificates that have been trusted in the global (Windows) certificate store will be listed.

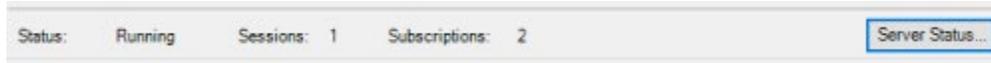
- d. Certificate Authorities - a listing of supported certificate authorities that issue third-party encryption certificates such as DigiCert, VeriSign, GlobalSign, Symantec and more.
  - e. Temporary Certificates - a listing of certificates that have been trusted just for this session only - such as client certificates where the certificate was accepted by DataHub even though it was invalid for some reason due to one of the previously covered settings being enabled. Such certificates, unless they're Accepted, will be removed when DataHub is restarted and the UA client will have exchange certificates again.
8. **Application Certificate** - clicking this button will display the properties of the currently assigned DataHub certificate (a self-signed certificate is generated by DataHub upon install).



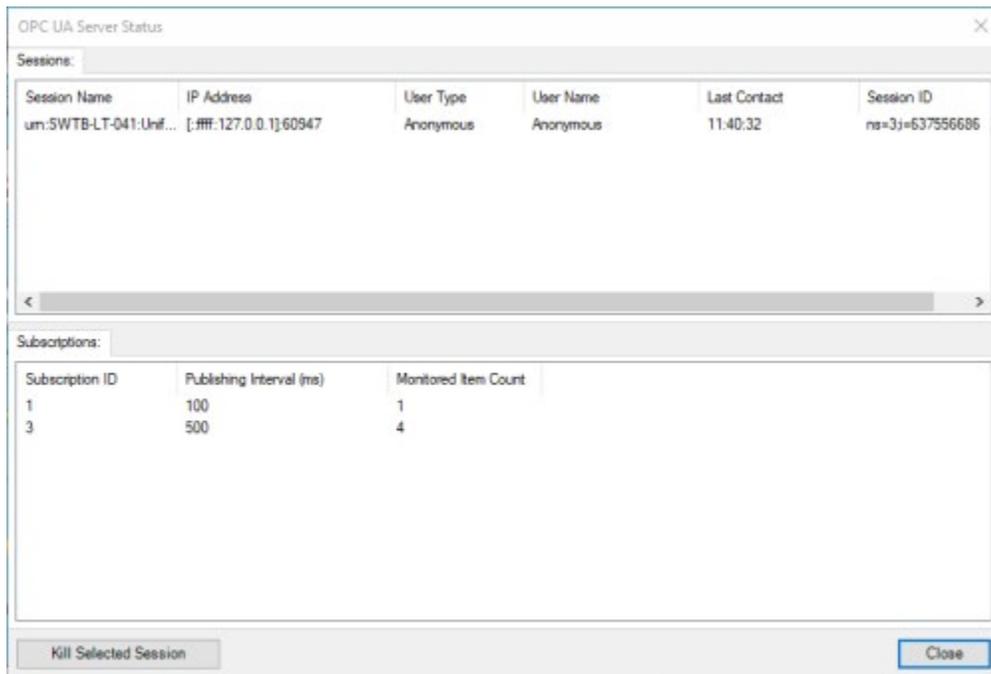
Additionally, you have options here to **Export** this certificate (in the event you need to manually exchange certificates and need to import DataHub's certificate into your OPC UA client application (consult your client's documentation for managing certificates).

You can also **Assign** a different certificate such as one from a third-party certificate authority for DataHub. Or you can **Regenerate** the self-signed DataHub certificate, which is useful if there is a chance the certificate has been compromised, if the computer name changes or if the current certificate has expired.

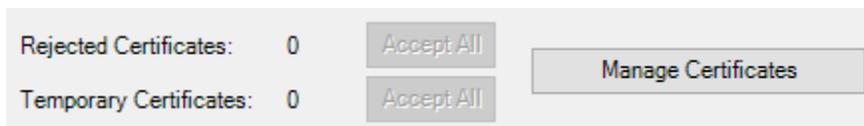
Back in the main DataHub OPC UA settings section, you will also notice a Status bar, which provides a high-level overview of whether or not the DataHub UA server is currently running or not, how many sessions from UA clients currently exist and how many current subscriptions are active.



Clicking the "Server Status" button provides a more detailed view including specifics of connected UA clients including the IP Address, user authentication details and more. You can even use the "Kill Selected Session" to discontinue the session of a connected UA client by highlighting it and clicking the button.

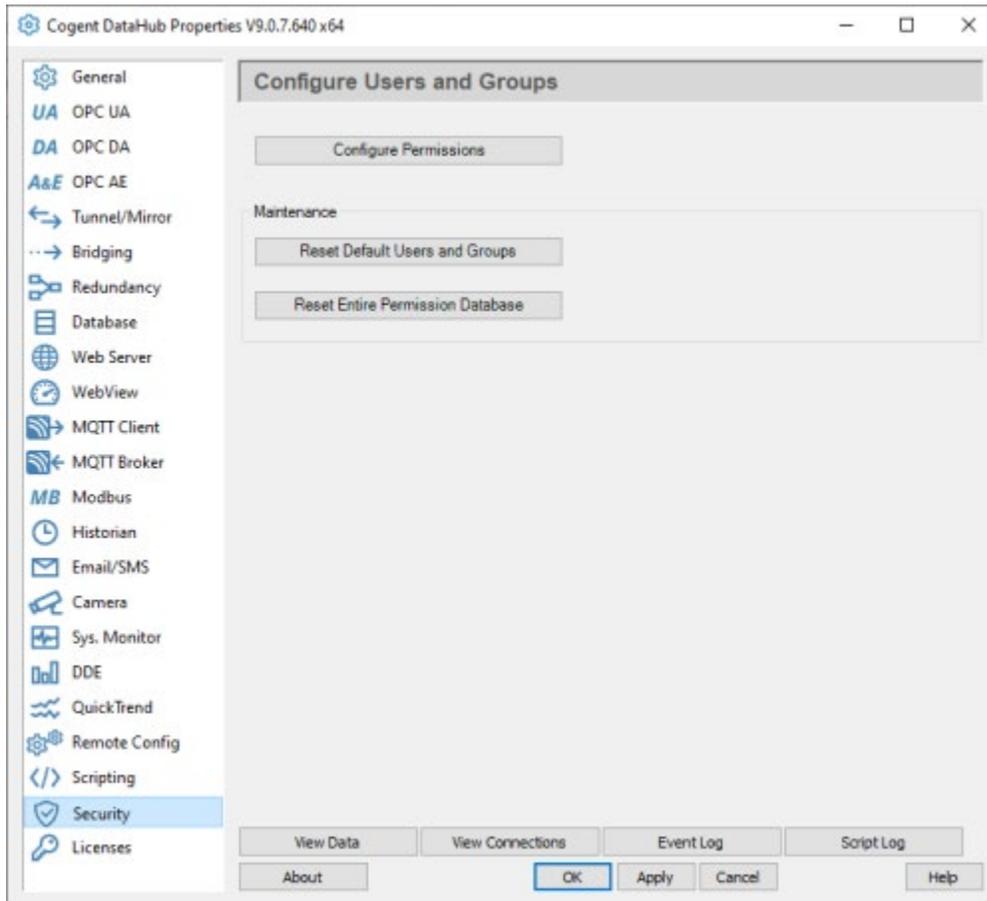


And another convenient features for managing OPC UA client certificates of clients attempting connections to DataHub are the **Rejected Certificates** and **Temporary Certificates** counters with their corresponding "Accept All" buttons. While not recommended on a live production system, for testing purposes, these options make it easy to get proof-of-concepts up and running by blanket accepting any and all OPC UA client certificates that have been exchanged but not trusted/accepted yet.



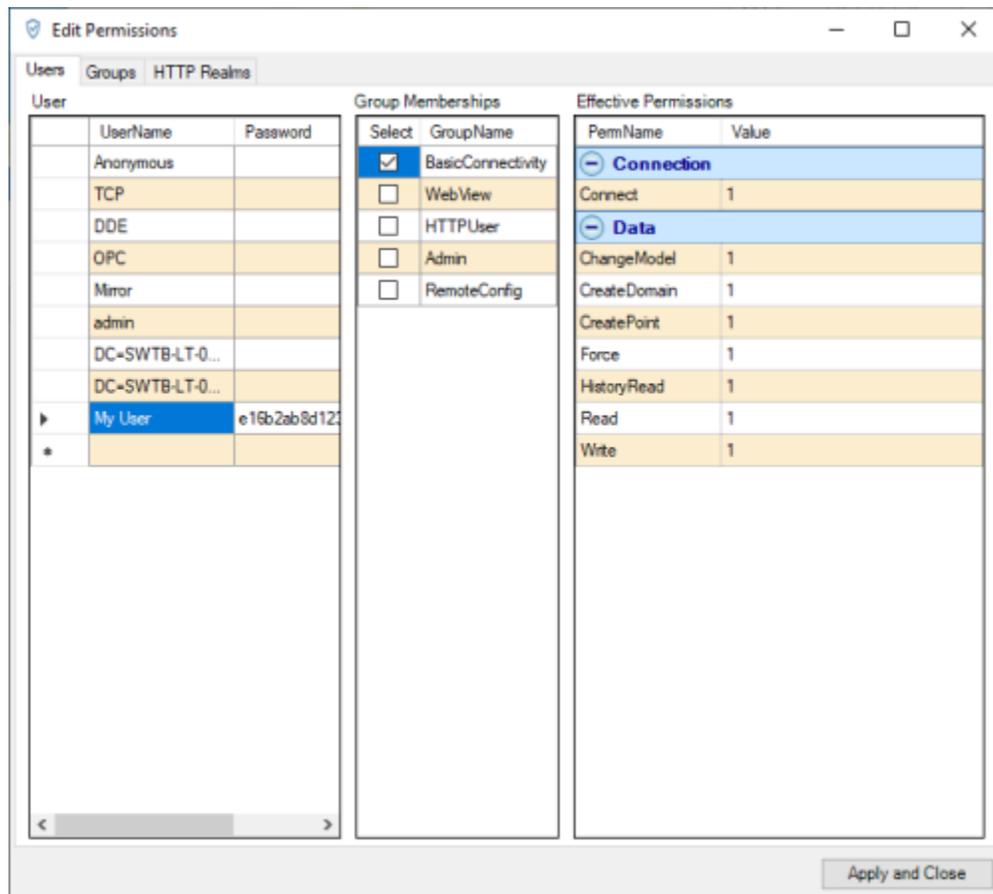
## 2. DataHub Security Settings

DataHub has built-in authentication and user permissions management in the "Security" - this is where the user an OPC UA client passes for UserName authentication must exist with valid permissions and password for DataHub to accept the connection. Just click the "Configure Permissions" button to get started.



Permissions are defined at the group level so that means the user can belong to any defined group as long as that group allows the user the required access to the nodes/points you need to read and/or write in DataHub. At a minimum, your user should belong to the "Basic Connectivity" group to have the appropriate permissions for connect, read and write.

It's simple to add a new user - just click in the empty space at the bottom of the list in the Users section and enter the UserName and then a corresponding Password. Then check the box of the desired Group Membership (selecting "BasicConnectivity" at a minimum).



Just make note of the UserName and Password you define here since you will need to enter it in your OPC UA client application to successfully connect to DataHub (if you're using username/password authentication).

## Connecting Your OPC UA Client to Cogent DataHub

Using the information we've just discussed, you can get your OPC UA client connected to DataHub You'll want to step through the following list, as a general rule (or [watch our tutorial video on connecting an OPC UA Client to DataHub here](#)):

1. In the DataHub OPC UA section:
  - a. Enable "Act as an OPC UA Server"
  - b. Assuming that you'll use the default port number for the opc.tcp endpoint, make note of the port which defaults to Port 51310 for use in your client. (Alternately, change the Port, as required, and make note of that number. )

- c. By default, DataHub is configured to make OPC UA connections as easy as possible - as such, it shouldn't be necessary to make any changes in the "Advanced" settings, since all security and authentication options are supported by default.
  - d. Export the security instance certificate from your OPC UA client (how to do this varies by client application – [please consult the help documentation for your OPC UA client for details on exporting its OPC UA security instance certificate](#)).
  - e. Go into "Manage Certificates" and select either the "OPC UA Private" or "OPC UA Global" certificate store option and click Import and browse to the OPC UA client certificate from the previous step.
  - f. Export DataHub's security instance certificate by clicking the "Application Certificate" button and clicking the "Export" button.
  - g. Import DataHub's security instance certificate into your OPC UA client (how to do this varies by client application – [please consult the help documentation for your OPC UA client for details on importing server security instance certificates and trusting them](#)).
  - h. Configure a user with the required permissions in the DataHub Security section ensure it belongs to the "BasicConnectivity" group.
  - i. Back in the OPC UA setting, highlight the opc.tcp endpoint and click the "Copy Endpoint to Clipboard" button to copy the fully-qualified endpoint URL (either for pasting directly into your OPC UA client (if installed locally) or for reference.
  - j. Click Apply at the bottom of your DataHub configuration.
2. Configure a connection from your OPC UA client using the endpoint address:
    - a. You'll need the URL you just copied from DataHub.
    - b. You'll need to confirm which of the security options DataHub supports is also supported by your OPC UA client (remember, DataHub supports Basic256Sha256, Basic256, Basic128Rsa15, None).
    - c. You'll need the username and password that you configured in the DataHub Security section (if applicable – by default, Anonymous login is enabled so if you didn't change that in DataHub, you can leave the username and password blank in your UA client).
    - d. [Consult your client's help documentation on the specific steps required to complete the connection and how to either browse for nodes/tags in DataHub or how to manually add nodes/tags.](#)

For a walkthrough of connecting an OPC UA client application to DataHub, [watch the detailed how-to video here](#).

## Connecting Cogent DataHub to Another OPC UA Server

Again using the information we discussed earlier, you can get DataHub connected to your OPC UA server. You'll want to step through the following list, as a general rule (or [watch a tutorial video that covers connecting DataHub to other OPC UA servers here](#)):

1. Make sure your other OPC UA server is properly configured to accept OPC UA client connections including enabling the interface (if applicable), having your OPC UA endpoint configured and any username/password authentication setup properly (consult the help documentation for your other OPC UA server for details on preparing for OPC UA clients to connect including how to export the security instance certificate).
2. You'll need the following details from your other OPC UA server in order to configure the DataHub to connect:
  - a. OPC UA endpoint URL (including Port)
  - b. Security policies that are supported and enabled (including whether Sign and/or Sign & Encrypt are required)
  - c. The security instance certificate from that OPC UA server.
3. Import the security instance certificate from your other OPC UA server in DataHub by clicking "Manage Certificates" and selecting either the "OPC UA Private" or "OPC UA Global" certificate store and clicking the Import button, then browse to the certificate file.
4. Export DataHub's security instance certificate by clicking the "Application Certificate" button and clicking the "Export" button.
5. Import DataHub's security instance certificate into your OPC UA server (how to do this varies by server application – please consult the help documentation for your OPC UA client for details on importing server security instance certificates and trusting them).
6. In the OPC UA section of DataHub, make sure "Act as an OPC UA Client to these servers" is enabled.
7. Click the "Add" button to start configuring a new OPC UA client connection.
  - a. Enter a meaningful "Connection Name".
  - b. If the OPC UA server is local, it might be browsable from the "Endpoint URL" dropdown - if not, you can also paste or type in the endpoint URL from your OPC UA server in the "Endpoint URL" field.
  - c. Select a "Security Policy" appropriate for your OPC UA server (remember, DataHub supports Basic256Sha256, Basic256, Basic128Rsa15, None so select one of those options based on what is enabled and supported in your OPC UA server).

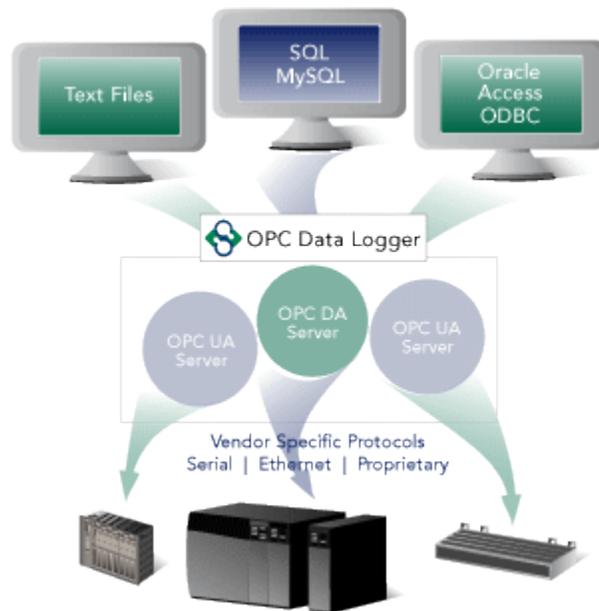
- d. Select the desired "User Token Type" based on what your OPC UA server supports and is enabled (not all OPC UA servers support certificate based authentication or anonymous log-in so confirm what options are required by your server and make sure what you configure here matches - including the UserName and Password fields, if you select UserName for the type).
  - e. Click "Connection Test" to confirm that the above settings are correct (if not and you receive an error, make the appropriate changes based on the warning/error message).
  - f. I would recommend keeping the defaults for the "Data Transfer" section settings, unless you have a specific reason to change them (such as you know your OPC UA server doesn't support a Subscription read method, for instance).
  - g. Click the "Configure Nodes" button and browse your OPC UA server address space and check the box next to the desired nodes/points that you want to access and click OK.
  - h. Enter a meaningful "Data Domain Name" for where your selected nodes/points will be stored.
  - i. Click OK.
  - j. Click Apply at the bottom of the DataHub configuration.
8. Once your OPC UA Client settings for your connection to the underlying OPC UA server are complete, you can quickly test your configuration using the View Data button at the bottom of the DataHub configuration window (find your Data Domain and expand it until you find the branch containing the nodes/points you browsed for and selected). You should be able to confirm there is good quality and changing data here.

For a walkthrough of connecting DataHub to another OPC UA server, [watch the detailed how-to video here](#). As always, you can try out everything we've just covered yourself with the [free trial of DataHub](#).

And another important resource is the [Cogent DataHub Video Resources web page](#). It contains a number of detailed how-to videos on a range of topics related to configuring Cogent DataHub.

## OPC Data Logger OPC UA Security (Client)

[OPC Data Logger](#) is designed as an effective solution for reliable, event-driven logging of data from OPC server data sources (both OPC UA and OPC DA) to SQL and ODBC databases (including Microsoft Azure SQL) or text and CSV files. The wizard-based interface creates a flexible configuration which can be scaled with ease. And OPC Data Logger is OPC Foundation lab-certified to ensure compatible with OPC certified servers.

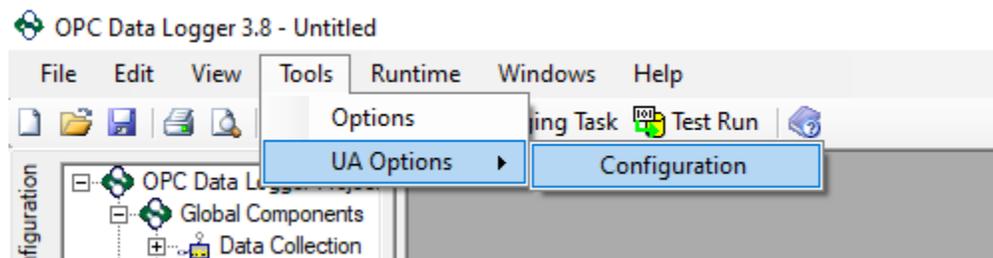


## OPC UA Configuration Components in OPC Data Logger

The settings relevant to OPC UA in OPC Data Logger can be found in several different locations:

### 1. *DataHub OPC UA Client Settings*

Accessible under the Tools -> UA Options -> Configuration menu in the OPC Data Logger Configuration window, this is where security instance certificates are managed for the OPC Data Logger and for OPC UA servers, as well as, optional registration with an OPC UA discovery server.



The following sections are available:



1. **Trusted Server** - This section is where certificates are managed for the OPC UA server or servers that you wish to log data from. In order for OPC Data Logger to connect to an OPC UA server, the server must trust OPC Data Logger and OPC Data Logger must trust the OPC UA server. This is accomplished by exchanging security instance certificates.

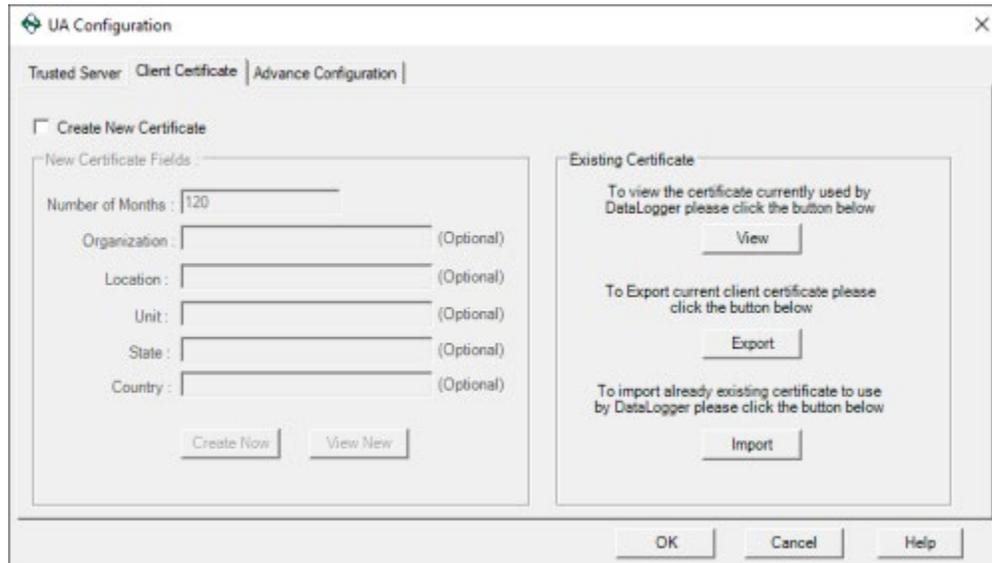
If you have attempted to connect to an OPC UA server without first exchanging security instance certificates, the connection will likely fail and you will find the certificate of the server located in the Rejected Certificates section you see here. To trust that server, simply highlight its certificate and click the right >> button to move it to the Trusted Certificates section.

Alternately, if you have previously trusted an OPC UA server but wish to revoke that trust, simply find its certificate in the Trusted Certificates section and click the left << button to move it to the Rejected Certificates section.

You can also export your OPC UA server certificate (consult the documentation for your OPC UA server for details on exporting the security instance certificate) and use the **Import** button here to trust the server in advance of attempting a connection. Just click the Import button and browse to the certificate file (with a .der or .cer extension).

You can also easily delete any trusted or rejected certificates here as needed.

2. **Client Certificate** - This section is where you manage OPC Data Logger's own security instance certificate.

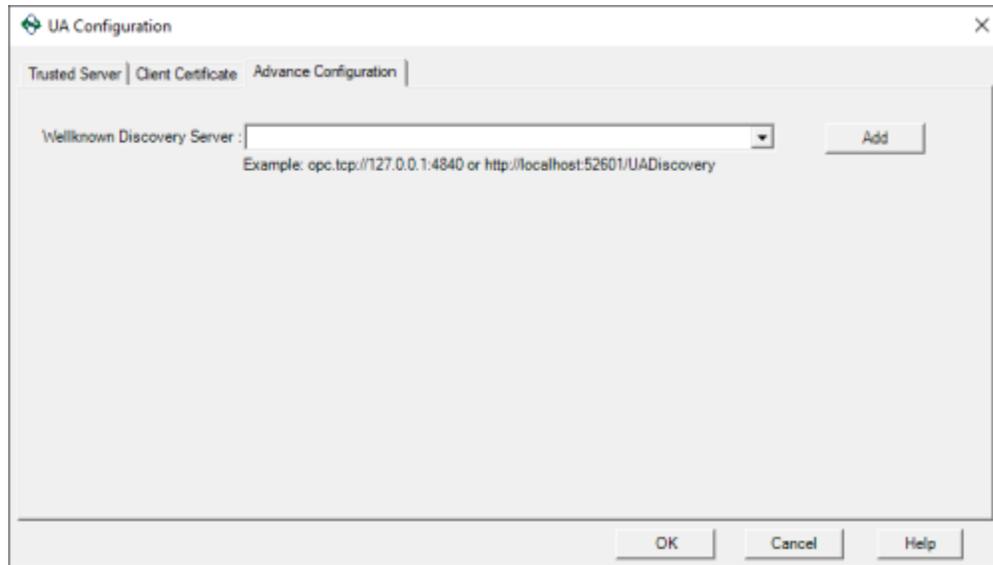


By default, OPC Data Logger generates a self-signed certificate upon install. You can **View** the details for the certificate currently issued, you can **Export** the certificate and then import it into your OPC UA server (consult the documentation for your OPC UA server for details on importing security instance certificates from UA clients) or you can **Import** certificates from third-party certificate authorities (such as Thawte, Verisign, etc).

You can also reissue the self-signed certificate using the **Create New Certificate** option and entering the available fields including the valid duration before the certificate expires and your organization name and location details.

3. **Advance Configuration** - And, while OPC Data Logger doesn't install with one, the Advance Configuration section is available to define any Local Discovery Server (LDS) services that you may have installed either on the same machine as OPC Data Logger or on another machine that is network accessible by OPC Data Logger. This includes the Local Discovery Server available from the [OPC Foundation \(available to registered users\)](#).

A discovery server for OPC UA is a dedicated service with which an OPC UA server can register, allowing OPC UA clients like OPC Data Logger to then look at the LDS to “browse” for available OPC UA servers they can connect to.



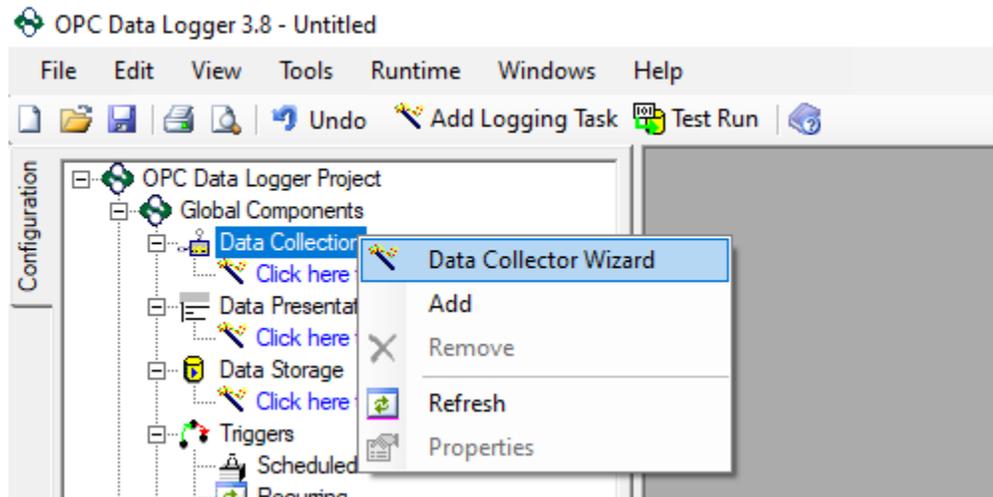
Again, for those familiar with OPC DA Classic, this is similar to what you might be used to with OPCEnum for browsing OPC DA servers.

You can either select an existing Discovery URL from the dropdown, if available, or manually enter the Local Discovery Server's endpoint URL (such as `opc.tcp://127.0.0.1:4840` for a local server) and click the **Add** button which will allow you to browse the discovery server when defining an OPC UA Data Collector in OPC Data Logger.

## 2. OPC Data Logger Data Collector Wizard

The settings for connecting to your OPC UA server in OPC Data Logger such as the endpoint and security settings are defined in the Data Collector configuration, which is wizard driven (you can also manually add Data Collectors but using the wizard is the recommended method).

Simply right-click on the Data Collection section of the tree view under Global Components and select Data Collector Wizard to launch the wizard.



The wizard steps you through the entire configuration in the following key steps:

1. **Selecting the Data Collector Type** - Here you simply select OPC Unified Architecture (UA) Interface from the Type dropdown.



- Naming Your Data Collector** - Here you give the new Data Collector a meaningful or "friendly" name that represents it in the OPC Data Collector, such as the name of the OPC UA server you'll be connecting to.

Add DATA SOURCE Wizard

**OPC DATA LOGGER 3.8** DATA COLLECTION WIZARD

Enter a name for the new Data Collector.

TOPServerOPCUA

This is simply a "Friendly Name" for this data collector.

Cancel < Previous Next >

- Configuring the OPC UA Endpoint, Security and Authentication Options** - Here is the meat of the OPC UA configuration as it pertains to your specific OPC UA server.

Add DATA SOURCE Wizard

**OPC DATA LOGGER 3.8** DATA COLLECTION WIZARD

Discovery Url: (Optional) [ ] Get Server Endpoint

Server Url: opc.tcp://127.0.0.1:49380 Get Security Modes

Security: SignAndEncrypt - Basic256Sha256

User Authentication

Use Authentication

Username: johnsmith

Password: [ ]

Configuration

Show Configuration [ ? ]

Cancel < Previous Next >

If a Local Discovery server is available, you can select it from the dropdown and click "Get Server Endpoint". Otherwise, assuming a discovery server is not available, you can simply enter the endpoint from your OPC UA server here and click "Get Security Modes".

You can then select the desired level of encryption (including None, if so desired) from the "Security" dropdown, which will only display security policies enabled and supported by your OPC UA server.

Additionally, if you want to use user authentication to access your OPC UA server (or if your OPC UA server doesn't support anonymous log-in) you will enable "Use Authentication" here and enter a valid Username and Password as defined in your OPC UA server. To use anonymous log-in, if your UA server supports it and you don't wish to use authentication, simply leave "Use Authentication" disabled here.

Also, the "Show Configuration" button is another method to access the UA Configuration settings that we covered previously.

## Connecting OPC Data Logger to Your OPC UA Server

Using the information we've just discussed, you can get OPC Data Logger configured to connect to your OPC UA server. You'll want to step through the following list, as a general rule (or [watch our tutorial video on connecting OPC Data Logger to TOP Server here for an example](#)):

1. Make sure your other OPC UA server is properly configured to accept OPC UA client connections including enabling the interface (if applicable), having your OPC UA endpoint configured and any username/password authentication setup properly (consult the help documentation for your other OPC UA server for details on preparing for OPC UA clients to connect including how to export the security instance certificate).
2. You'll need the following details from your other OPC UA server in order to configure the OPC Data Logger to connect:
  - a. OPC UA endpoint URL (including Port)
  - b. The security instance certificate from that OPC UA server.
3. Import the security instance certificate from your other OPC UA server in the OPC Data Logger UA Configuration under "Trusted Server" by clicking the "Import" button and browsing to the certificate file from Step 2 above.
4. Export the security instance certificate for the OPC Data Logger in the UA Configuration under "Client Certificate" by clicking the "Export" button.
5. Import the OPC Data Logger security instance certificate into your OPC UA server and trust it ([consult the help documentation for your other OPC UA server to determine how to import and trust client certificates](#)).

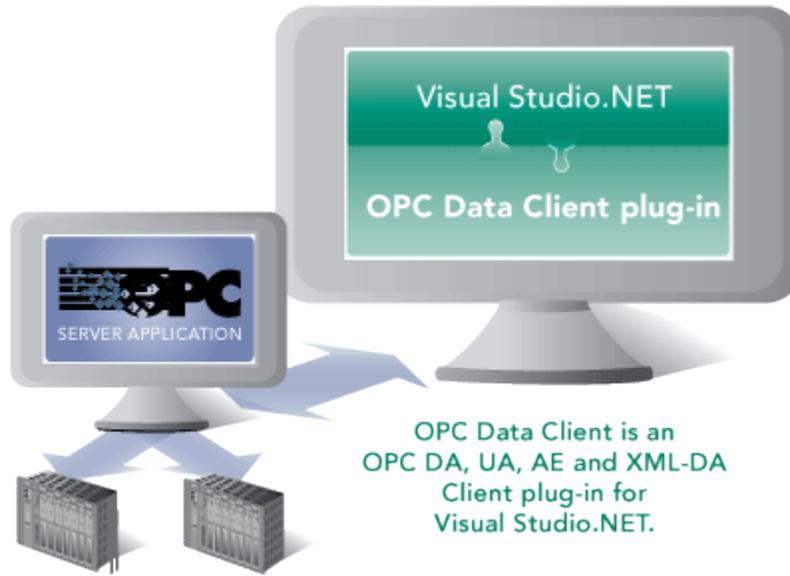
6. Configure a new Data Collector by right-clicking on "Data Collection" and selecting "Data Collector Wizard" and step through the wizard:
  - a. Select "OPC Unified Architecture (UA) Interface" from the Data Collector type.
  - b. Give the new Data Collector a meaningful name.
  - c. If a Discovery Server is available, select it from the dropdown. Otherwise, enter the endpoint URL of your OPC UA server in the "Server Url" field and click "Get Security Modes".
  - d. Select the desired security encryption method from the "Security" dropdown.
  - e. If using username/password authentication (leave unchecked if your UA server supports anonymous login and you plan to not use authentication), enable "Use Authentication" and enter a valid Username and Password for your UA server (consult your UA server documentation for details on defining user security).
  - f. Complete the new Data Collector by finishing the wizard.
7. The remainder of the configuration is specific to configuration of data presentations, data storage and other non-OPC UA related configuration. [For how-to videos on logging to text files and databases, click here.](#)

For a walkthrough of connecting OPC Data Logger via OPC UA to TOP Server, [watch the detailed how-to video here](#). As always, you can try out everything we've just covered yourself with the [free trial of OPC Data Logger](#).

And another important resource is the [OPC Data Logger Video Resources web page](#). It contains a number of detailed how-to videos on a range of topics related to configuring OPC Data Logger.

## OPC Data Client OPC UA Security (Client)

[OPC Data Client](#) is a rapid development toolkit for quickly adding OPC client connectivity to custom applications including OPC UA and OPC Classic (DA, AE and XML-DA). The OPC Data Client is regularly tested at the OPC Foundation's OPC Interoperability Workshops and has passed their independent lab testing process, ensuring applications developed using OPC Data Client are compatible with OPC compliant solutions.



Among the many benefits of using OPC UA is the focus on security. Not only can the traffic be encrypted, but both the client and server endpoints can be forced to identify themselves (using TLS/SSL certificates), thereby preventing unauthorized client applications and/or machines from connecting to an OPC UA Server. This post focuses primarily on how to properly exchange and trust security certificates between OPC Data Client and the OPC UA server it will be connecting to.

### OPC UA Configuration Components in OPC Data Client

While this section of the e-book will go into the most detail on certificate exchange (since much of the other OPC UA configuration can be different depending on how the developer implements OPC Data Client in the custom application), we will also briefly discuss how OPC Data Client and the methods it exposes for handling OPC UA Server endpoints, encryption and user authentication.

#### ***1. OPC UA Server Endpoints in OPC Data Client***

An OPC UA Server Endpoint is a physical address available on a network that allows clients to access one or more services provided by a server. Server endpoint is specified by its URL string. For those more familiar with OPC DA Classic, the UA server endpoint is similar to an OPC DA Server ProgID.

OPC Data Client uses an [OPC UA Endpoint Selection Policy](#) that allows you to specify flexible criteria that OPC Data Client will use when picking up the endpoint from those made available by the target OPC UA server. The goal of this policy is to make client applications written with OPC Data Client as flexible as possible with respect to successful connections now and in the future, when the details for the server may have changed. For instance, this could happen if the currently selected encryption algorithm became outdated and the UA server stopped supporting it.

The developer might choose to expose fields for the user to define the endpoint and encryption to be used for a connection in a user interface or not - so how those settings are configured are really specific to how the developer implement OPC Data Client in the custom application.

```
using OpcLabs.EasyOpc.UA;
using System;
using System.Threading;

namespace UACertificateDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            EasyUAClient client = new EasyUAClient();

            try
            {
                var result = client.ReadValue(
                    "opc.tcp://192.168.111.55:49388",
                    "ns=2;s=Channel1.Device1.Tag1"
                );

                Console.WriteLine("Value: " + result);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
            }

            Thread.Sleep(30000);
        }
    }
}
```

## 2. OPC UA Authentication in OPC Data Client

In addition to encryption, an OPC UA connection with either be anonymous (not recommended for secure applications) or will require authentication between the UA client and server. OPC Data Client supports user authentications via username/password (most common), Kerberos token, security certificate or anonymous connections.

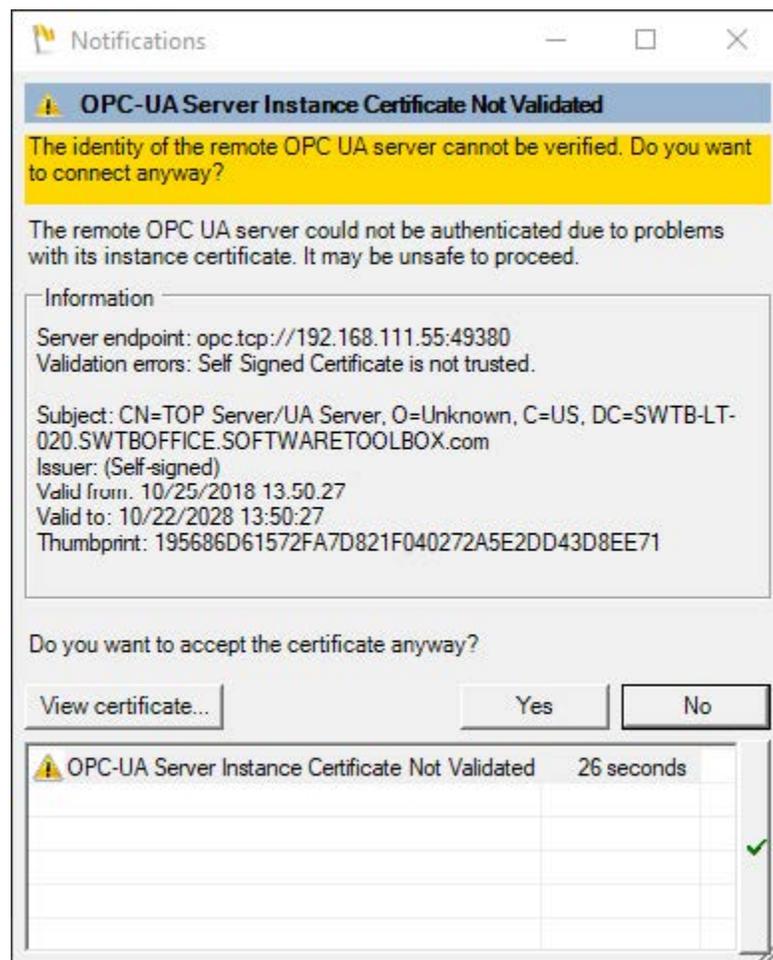
OPC Data Client provides [extension methods](#) that are part of the class for defining OPC UA endpoints that allow developers to handle authentication.

### 3. Exchanging OPC UA Security Certificates in OPC Data Client

For a custom OPC Data Client application to successfully connect to an OPC UA server that requires encryption (which is recommended for secure communications), it is necessary to exchange and trust security certificates with the OPC UA server.

#### 1. Exporting and Exchanging OPC Data Client certificate with OPC UA server

This can be accomplished most easily by attempting to make a connection from your client application to the underlying OPC UA server. You'll receive a notification similar to the following.



This indicates that the OPC UA server certificate is not trusted and prompts the user to choose whether the server certificate should be trusted. Selecting "No" will reject the OPC UA server certificate and the application will throw an exception.

In order to trust the certificate and continue with the connection process, the certificate must be trusted with the "Yes" button. The selection here is not persisted between runs;

this means the next time the app is launched this prompt will be presented again.

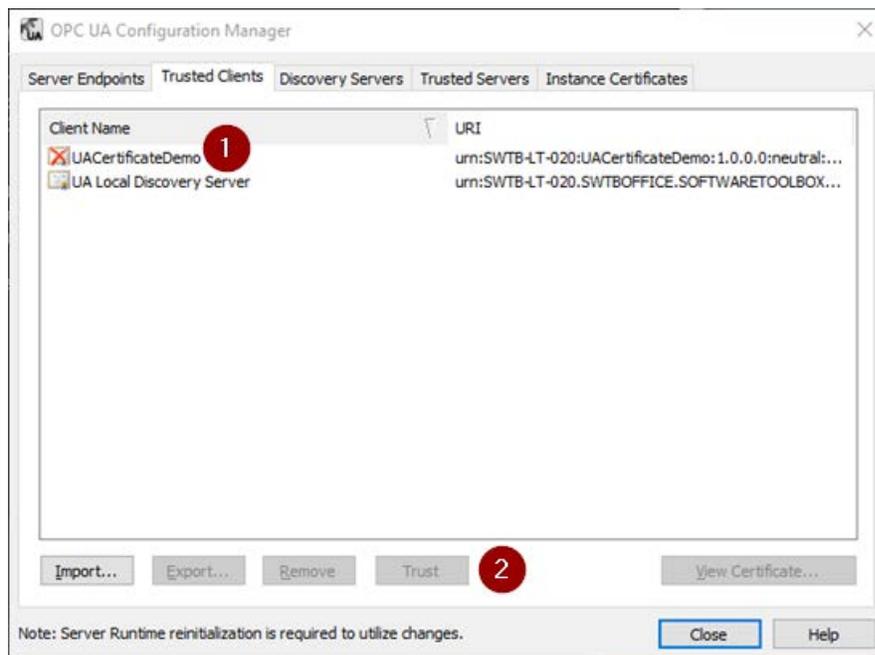
Now, even though the server certificate has been trusted by the client, an exception is thrown to the console (specifically a "BadSecurityChecksFailed" error). This is expected, at this point, given that the OPC UA server does not yet trust the client.

```
file:///C:/Users/mholbach/Dropbox/GitHub/UACertificateDemo/UACertificateDemo/bin/Deb
OpcLabs.EasyOpc.UA.OperationModel.UAException: An OPC-UA operation
rred, originating from '. The inner OPC-UA service exception wit
ails about the problem. ---> OpcLabs.EasyOpc.UA.UAServiceExceptio
ion. = BadSecurityChecksFailed.
---- SERVICE RESULT ----
StatusCode: {BadSecurityChecksFailed} = 0x80130000 (2148728832)
Description: Error establishing a connection.
```

## 2. Exporting and Exchanging OPC UA server certificate with OPC Data Client

In most OPC UA servers, a failed connection attempt by an OPC UA server will result in the client's certificate being listed in the UA server in it's OPC UA configuration settings as "untrusted", "rejected" or some other similar designation. So, after that initial failed connection, establishing that trust is as simple as finding that section in your OPC UA server and changing that designation to be trusted.

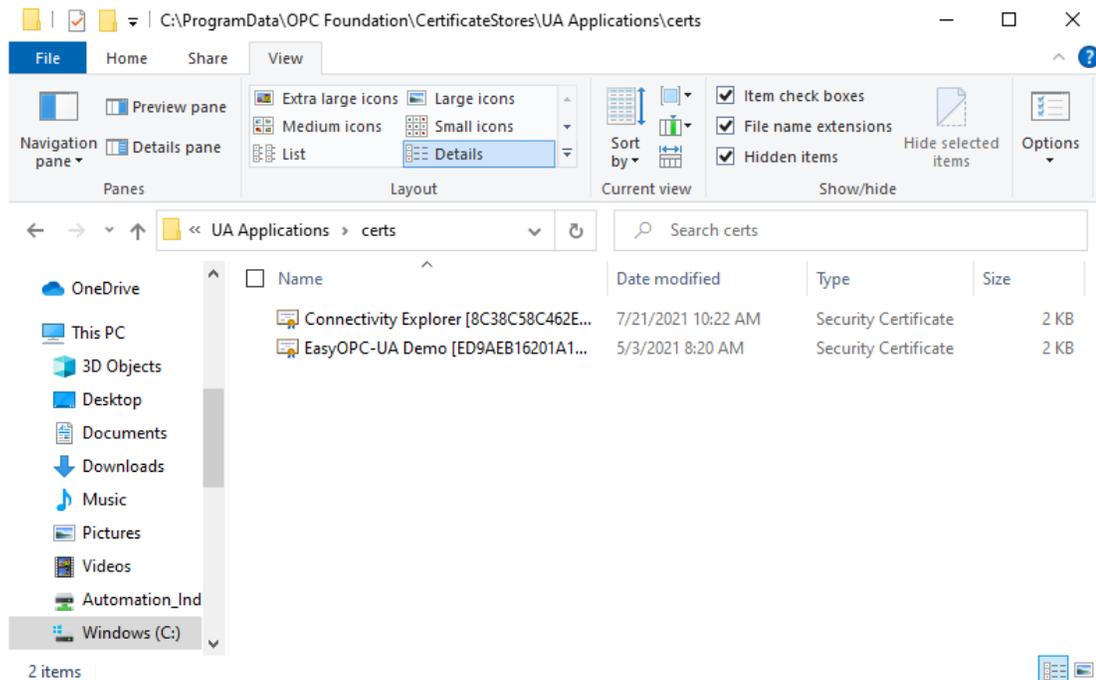
For example, see below how TOP Server handles OPC UA client requests prior to trust and how you can simply highlight the client's certificate (1) and click the "Trust" button (2).



You'll need to consult the help documentation for your OPC UA server to determine specifically how to import and trust client certificates (since this varies between OPC UA servers).

### 3. Exchanging OPC UA certificates manually

Now if you're working with an OPC UA server where the OPC UA client certificate does not automatically appear in a "rejected" state as described above, the certificate will need to be manually imported into the server's trusted client certificate list.



The OPC Data Client's OPC UA Client certificate can be found by default at:

**%CommonApplicationData%\OPC Foundation\CertificateStores\UA Applications\certs**

The %CommonApplicationData% directory is most commonly located at `C:\ProgramData\` on most systems. Steps to import the certificate into an OPC UA server manually will vary based on the OPC UA Server in use. As mentioned above, you'll need to consult your OPC UA server's documentation for instructions since some servers provide a means to import certificates in the user interface and others require you to simply copy the certificate file into a specific directory.

And to manually trust your OPC UA server's certificate, consult the server's

documentation on either how to manually export its certificate or where the certificate is located and copy the certificate the same directory where your OPC Data Client certificate is located (as shown above).

## Get Started Building a Custom OPC UA Client with OPC Data Client

For a full walkthrough detailing how to build no code and low code OPC UA client applications with OPC Data Client, [watch the detailed how-to video here](#). The detailed on-demand training covers how to build a basic OPC UA client and get connected to an OPC UA server. And, you can try out everything as you follow along with the [free trial of OPC Data Client](#).

And another important resource is the [OPC Data Client Video Resources list](#). It contains a number of additional detailed how-to videos on a range of topics related to using the OPC Data Client.

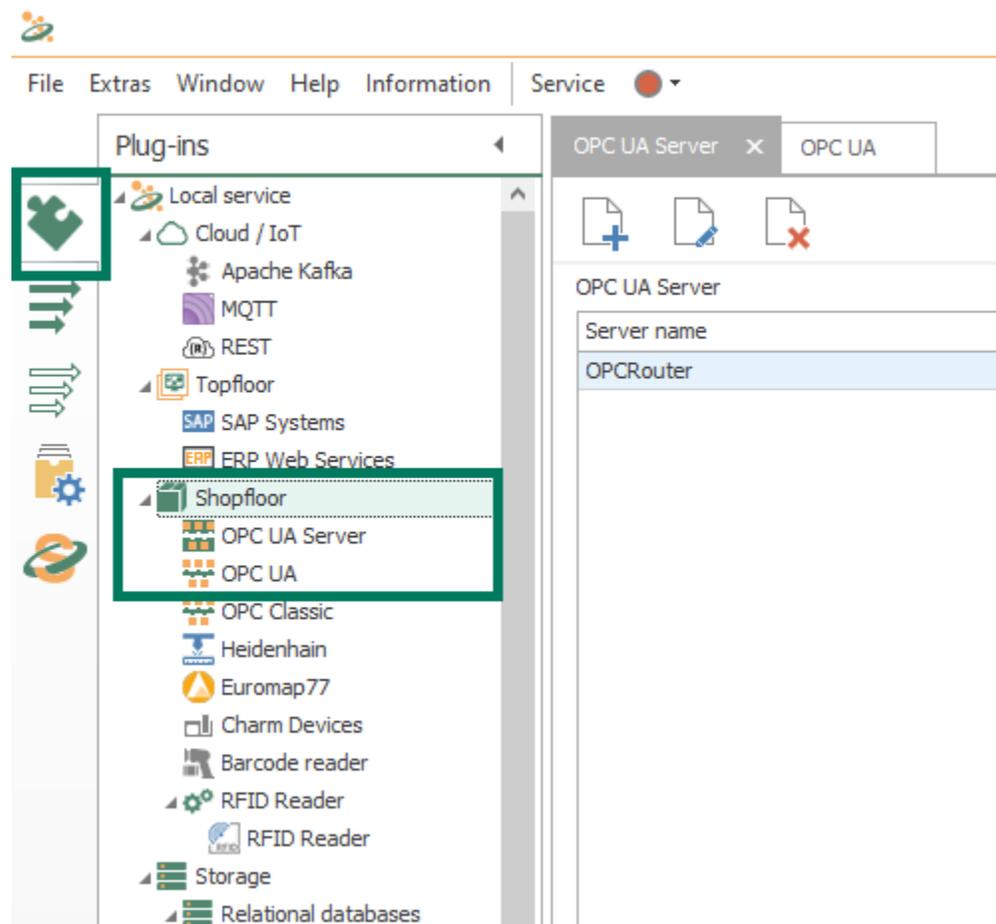
## OPC Router OPC UA Security (Client & Server)

OPC Router is a highly configurable software tool for integrating a wide variety of industrial, business, and IoT data sources using drag-and-drop visual workflows, reducing engineering time and risk in Industry 4.0, IIoT, and Digital Transformation applications.

In keeping with the many various functions available in [OPC Router](#), it can act as both an [OPC UA client and an OPC UA server](#). This, in conjunction with [OPC Router's other modules](#) and workflow capabilities, allows conversion to and from OPC UA for a variety of different systems that don't natively support OPC UA.

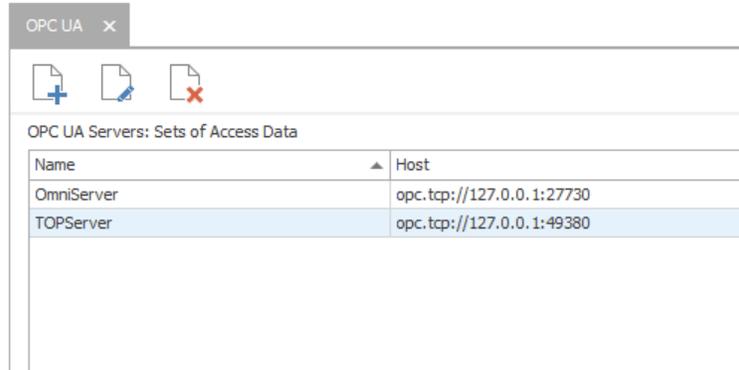
### OPC UA Configuration Components in OPC Router

The settings relevant to OPC UA in the OPC Router are located in the OPC UA plug-in settings for connecting to OPC UA servers and in the OPC UA Server plug-in settings for connections from other OPC UA clients. Both plug-ins are located under the "Shopfloor" branch of the "Plug-Ins" section as seen below.



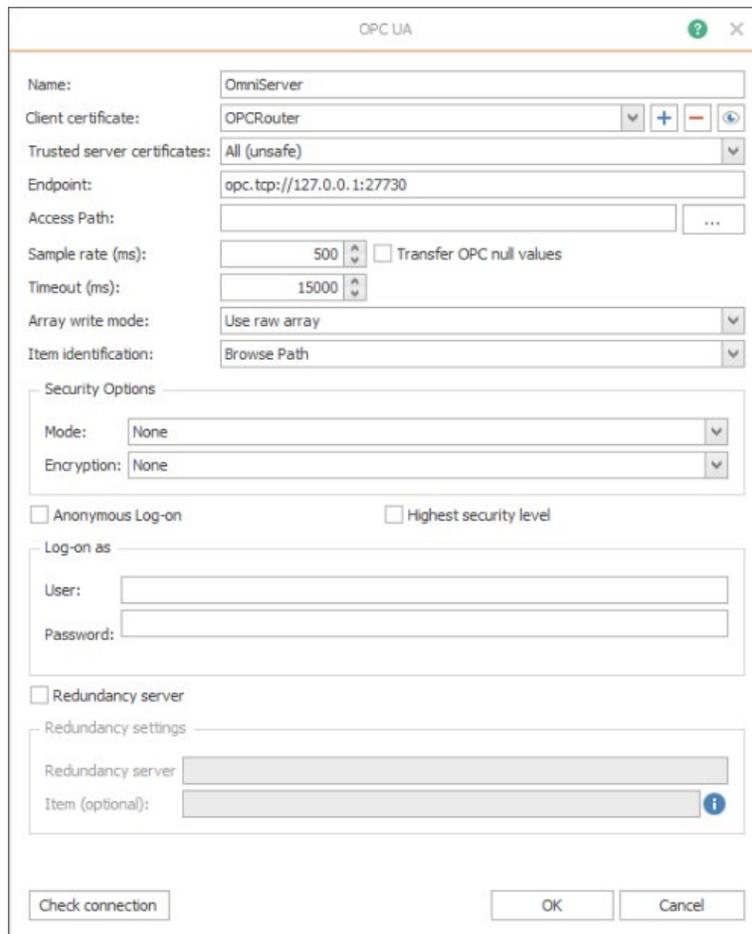
## 1. OPC Router OPC UA Plug-in Settings

The settings for connecting OPC Router to other OPC UA servers are configured via instances of the OPC UA plug-in.



Name	Host
OmniServer	opc.tcp://127.0.0.1:27730
TOPServer	opc.tcp://127.0.0.1:49380

Each instance represents a connection to an OPC UA server with its own settings for security parameters. Clicking "New" or "Edit" buttons presents the following available settings (this example is a local OPC UA connection to [OmniServer](#)):



OPC UA

Name: OmniServer

Client certificate: OPCRouter

Trusted server certificates: All (unsafe)

Endpoint: opc.tcp://127.0.0.1:27730

Access Path: ...

Sample rate (ms): 500  Transfer OPC null values

Timeout (ms): 15000

Array write mode: Use raw array

Item identification: Browse Path

Security Options

Mode: None

Encryption: None

Anonymous Log-on  Highest security level

Log-on as

User:

Password:

Redundancy server

Redundancy settings

Redundancy server:

Item (optional):

Check connection OK Cancel

Here are specific details on the key settings pertaining to security you will need to configure (the other settings shown above either don't pertain to security or can generally be left at the defaults):

1. **Name** - User-defined friendly name used by OPC Router to identify this connection.
2. **Client certificate** - Defines which of OPC Router's available security certificates to use for encryption for this specific connection. OPC Router's security certificates are managed through the Extras > Settings menu under Certificate Management (which will be covered in more detail shortly).

Alternately, you can:

- a. Click the "+" button to define a new self-signed security certificate for use with this connection
  - b. Click the "-" button to delete the selected security certificate.
  - c. Click the "eye" button to view the details of the selected security certificate.
3. **Trusted server certificate** - Selects which OPC UA server security certificates to accept/trust. The default setting is "All" which will result in the certificate of the OPC UA server being connected to getting trusted automatically. Other options include:
    - a. Router - if the certificate has not already been imported and trusted through [certificate management](#), you will be prompted to either trust or reject the OPC UA server's certificate when attempting to connect.
    - b. Windows - use this option if you've imported your OPC UA server's certificate to the "Third-Party Root Certification" section in the Local Machine Certificate Manager in Windows (Just search for "certificates" in Windows and select the options for "Manage computer certificates").
  4. **Endpoint** - For an OPC UA server, the endpoint is how an OPC UA client specifies a connection. For those familiar with OPC DA, this would be equivalent to the OPC DA Server ProgID at a very basic level. This setting is the URL for the OPC UA server you want to connect to - just manually enter the correct URL as determined in the configuration of your OPC UA server.

A server endpoint most commonly consists of the syntax "opc.tcp://" followed by either the IP address or Hostname of the machine where the OPC UA server is installed, followed by a colon and then the Port Number, which may or may not be configurable in the endpoint of the OPC UA server you're connecting to (consult the help documentation for your specific OPC UA server for details). Other possible valid prefixes in an endpoint URL are http:// or https://.

Additionally, the endpoint is where the level of secure encryption options that the OPC

UA client applications must support and use to make a connection to that endpoint. You'll need to confirm what sign and encrypt options are supported for your OPC UA server.

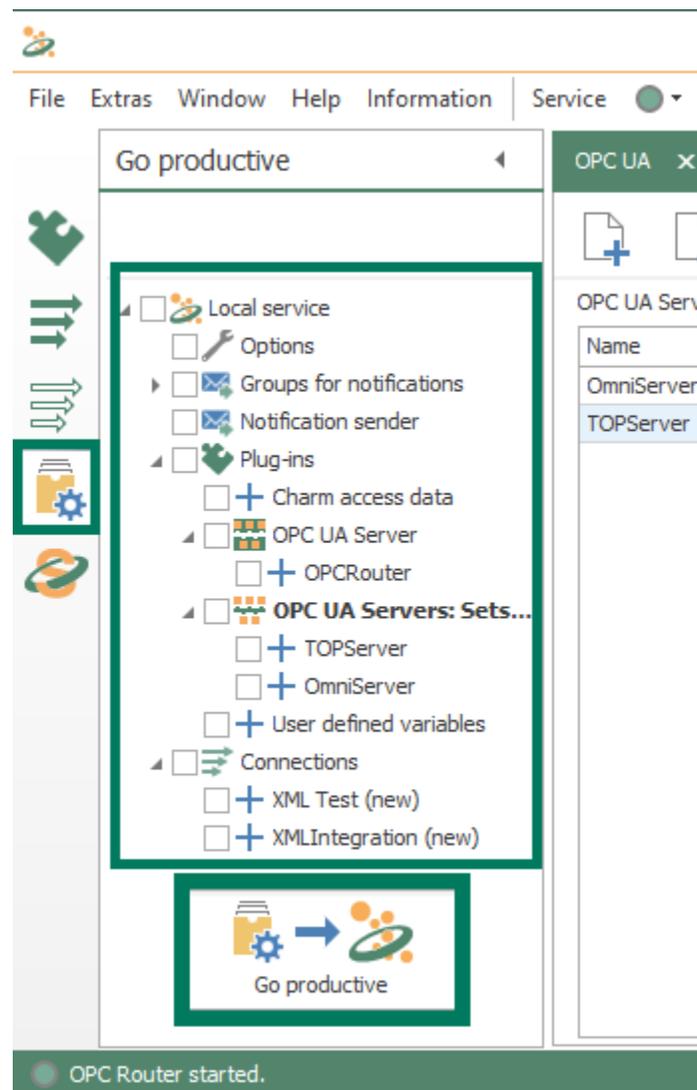
**NOTE:** If encryption is used for the connection, the Endpoint URL must be entered here exactly as it is specified in the server's certificate. For example, if the certificate contains "Server.domain.local" (i.e. a DNS name), the endpoint must also be addressed in this way, as opposed to using the IP address instead. Otherwise, the certificate will be rejected.

Generally, for each supported encryption option (other than None, of course), you can also select whether the endpoint requires Sign and/or Sign and Encrypt (for full details on Sign and Encrypt, refer back to the [Then What is Sign & Encrypt section](#) of this e-book).

5. **Security Options** - The following settings define which signing and encryption options will be used for this connection:
  - a. Highest security level - Enabled by default, this option will force the connection to use the most secure Mode and Encryption options supported by your OPC UA server for the connection. (Mode and Encryption settings are not available while this is enabled)
    - i. **NOTE:** If connection problems occur with this setting enabled, it is recommended to disable and set security options manually based on known supported settings for your OPC UA server endpoint.
  - b. Mode - Specifies whether the connection should use Sign, Sign and Encrypt or None (only available if "Highest security level" is disabled).
  - c. Encryption - Specifies which encryption algorithm to use for the connection - OPC Router currently support the following option in order from most to least secure: Basic256Sha256, Https, Basic 256, Basic128Rsa15, or None.
6. **Anonymous Log-on** - This setting (enabled by default) specifies that the OPC UA server doesn't require (or possibly support) user authentication via username and password. If your OPC UA server supports and requires user authentication, you will need to disable this setting, which will make the "Log-on as" authentication fields below it accessible.
7. **Log-on as** - The following fields are only available if "Anonymous Log-on" has been disabled:
  - a. User - Enter a username configured in your OPC UA server that is authorized to access the server via OPC UA.
    - i. **NOTE:** For details on user management and authentication in your OPC UA server, please consult the server documentation.

- b. Password - Enter the password associated with the username you specified above.
8. **Check connection** - Once you've configured the settings for your OPC UA server, you can click this button to initiate a test connection to the OPC UA server to confirm those settings are correct (if there are any issues, a meaningful message indicating what the problem was will appear in the Message section of the dialog).

Once you've edited the OPC UA settings for connecting to an OPC UA server, just click "OK" to save and exit the dialog. And, as with any other changes in OPC Router, make sure to go to the "Go Productive" section of the configuration and select the elements being used in your project and click the "Go productive" button to publish the changes.

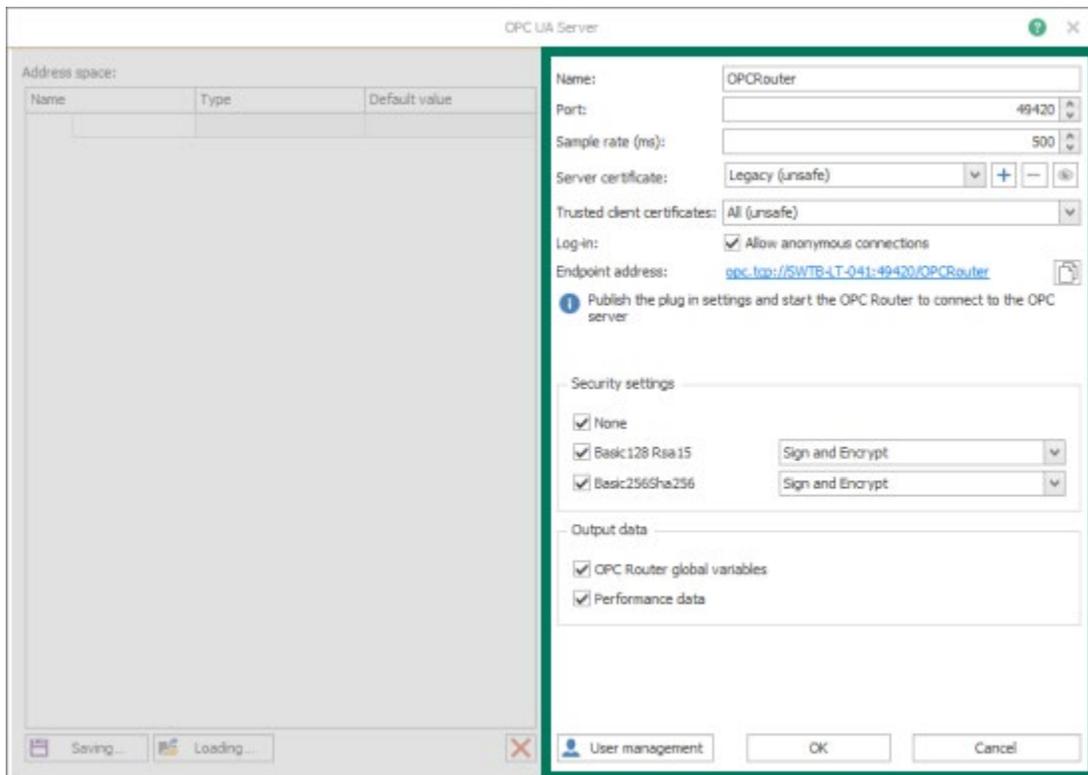


## 2. OPC Router OPC UA Server Plug-in Settings

The settings for connecting OPC UA clients to OPC Router are configured via instances of the OPC UA Server plug-in.



Each instance is a separate OPC UA server endpoint with its own specific port and security options (configuring the Address space, which defines data variables and objects that will be available to OPC UA client, is beyond the scope of this post. Clicking "New" or "Edit" buttons presents the following available settings:



Here are some details on the key settings pertaining to security you will need to configure for the OPC UA server endpoint (the other settings shown above either don't pertain to security or can generally be left at the defaults):

1. **Name** - user-defined friendly name used by OPC Router to identify this OPC UA server endpoint.
2. **Port** - the TCP port associated with this OPC UA server endpoint for OPC Router - this will be appended to the end of the endpoint URL.
3. **Server certificate** - this dropdown allows you to select a certificate (either a self-signed certificate that you've already create in OPC Router or a third-party certificate you've imported in Certificate Management (see Section [3. OPC Router Certificate Management](#) below)).

Alternately, you can:

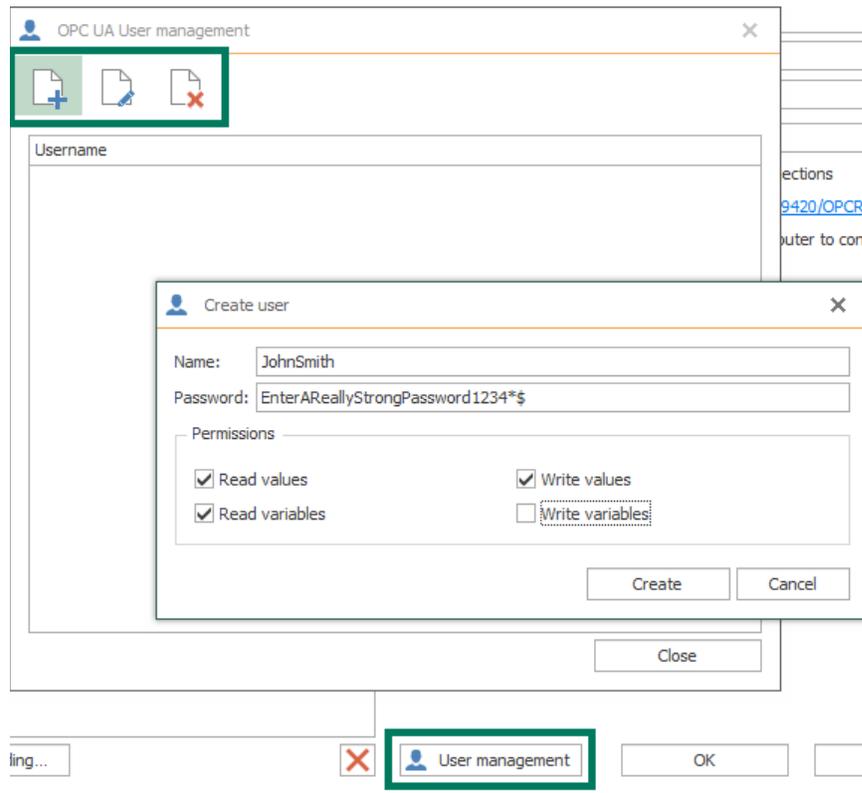
- a. Click the "+" button to define a new self-signed security certificate for use with this endpoint.
  - b. Click the "-" button to delete the selected security certificate.
  - c. Click the "eye" button to view the details of the selected security certificate.
4. **Trusted client certificates** - selects which OPC UA client security certificates to accept/trust for this endpoint. The default setting is "All" which will result in the certificate of any OPC UA client attempting a connection to this endpoint getting trusted automatically. Other options include:
    - a. Router - if the certificate has not already been imported and trusted through [certificate management](#), you will be prompted to either trust or reject the OPC UA client's certificate when attempting to connect.
    - b. Windows - use this option if you've imported your OPC UA client's certificate to the "Third-Party Root Certification" section in the Local Machine Certificate Manager in Windows (Just search for "certificates" in Windows and select the options for "Manage computer certificates").
  5. **Log-in: Allow anonymous connections** - enabled by default, this checkbox being enabled allows OPC UA clients to connect to this OPC Router endpoint without providing a username and password for authentication.

When disabled (recommended for the highest security), any OPC UA client connecting to this endpoint will need to provide a valid username and password as defined in the ["User management" configuration](#) for this endpoint.

6. **Endpoint address** - this is the actual endpoint you will need to specify in your OPC UA client for connecting to this endpoint and is based on the previous settings. Click the

"Copy" button to the right of the endpoint to copy it, making it easy to paste right into your OPC UA client (consult the documentation for your OPC UA client for how to specify the OPC UA server endpoint to connect to).

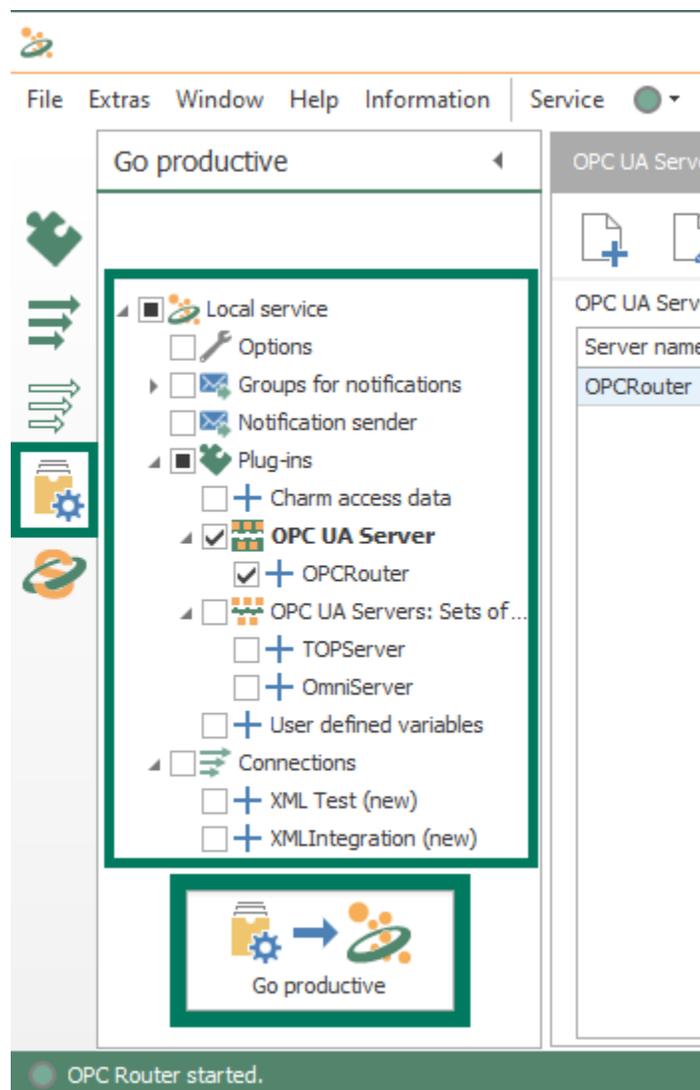
7. **Security settings** - these settings define which encryption and signing options will be available for connections to this specific OPC Router UA server endpoint from OPC UA clients. All options are enabled by default (**NOTE:** for highest security, disabling "None" is recommended).
  - a. None - least secure, no encryption will be used for UA connections using this option.
  - b. Basic128Rsa15 - available signing options include Sign and Encrypt (default and most secure), Sign or both can be available.
  - c. Basic256Sha256 - most secure. Available signing options include Sign and Encrypt (default and most secure), Sign or both can be available.
8. **User management** - clicking this button launches the OPC UA User management window which allows you to Add New, Edit or Delete users for this OPC UA endpoint for OPC Router.



- a. Name - the username that will need to be specified in your OPC UA client.

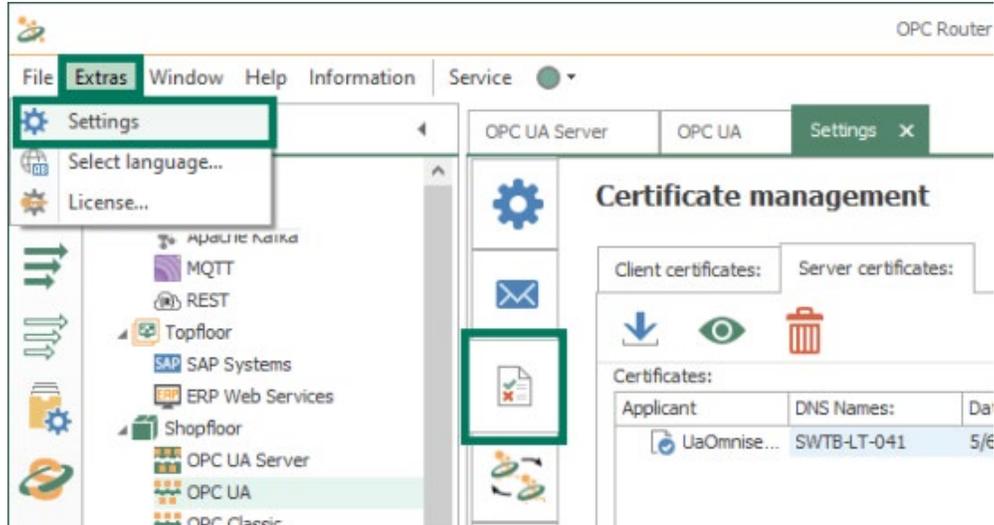
- b. Password - the password that will need to be specified in your OPC UA client (a longer password with a combination of mixed-case letters, numbers and special characters is recommended for the greatest security).
- c. Permissions - the capabilities an OPC UA client using this username/password will be entitled to. By default, the user is allowed to read values and variables and write values but you can use this to easily limit what an OPC UA client can and cannot do - for instance, you can allow read-only access.

Once you've edited the OPC UA Server settings for connections from an OPC UA client, just click "OK" to save and exit the dialog. And, as with any other changes in OPC Router, make sure to go to the "Go Productive" section of the configuration and select the elements being used in your project and click the "Go productive" button to publish the changes.

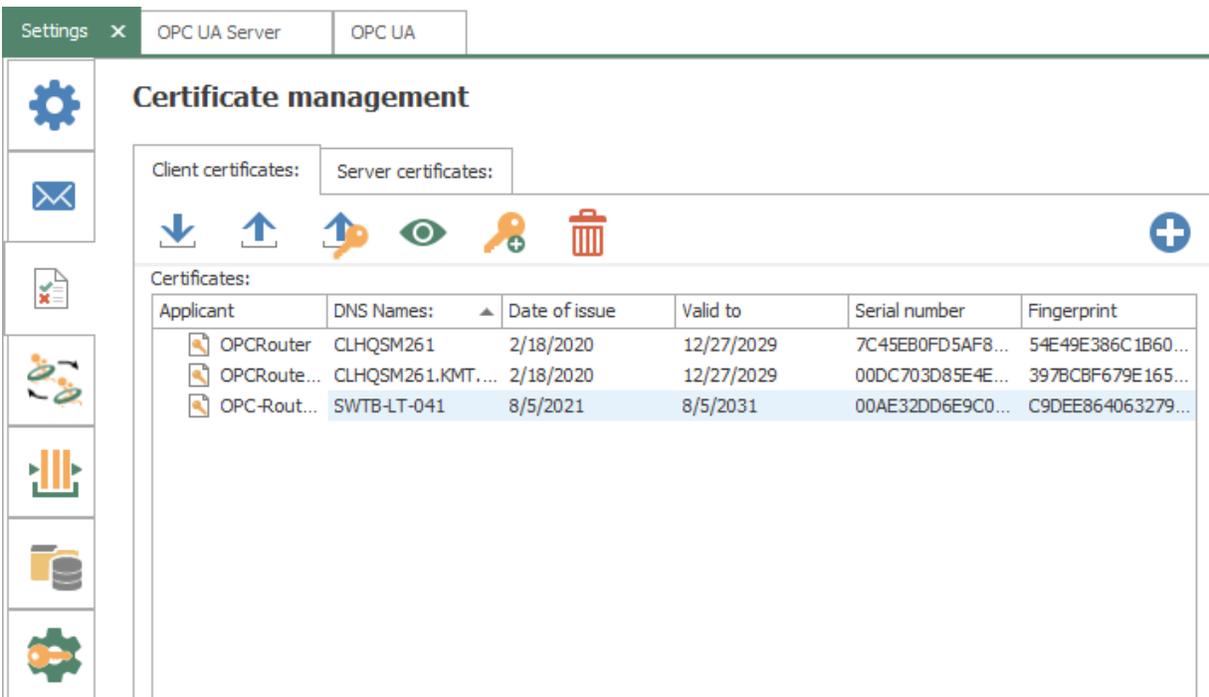


### 3. OPC Router Certificate Management

There is centralized certificate management for both OPC UA client and server interfaces for OPC Router via the **Extras > Settings** menu under the "Certificate management" section.



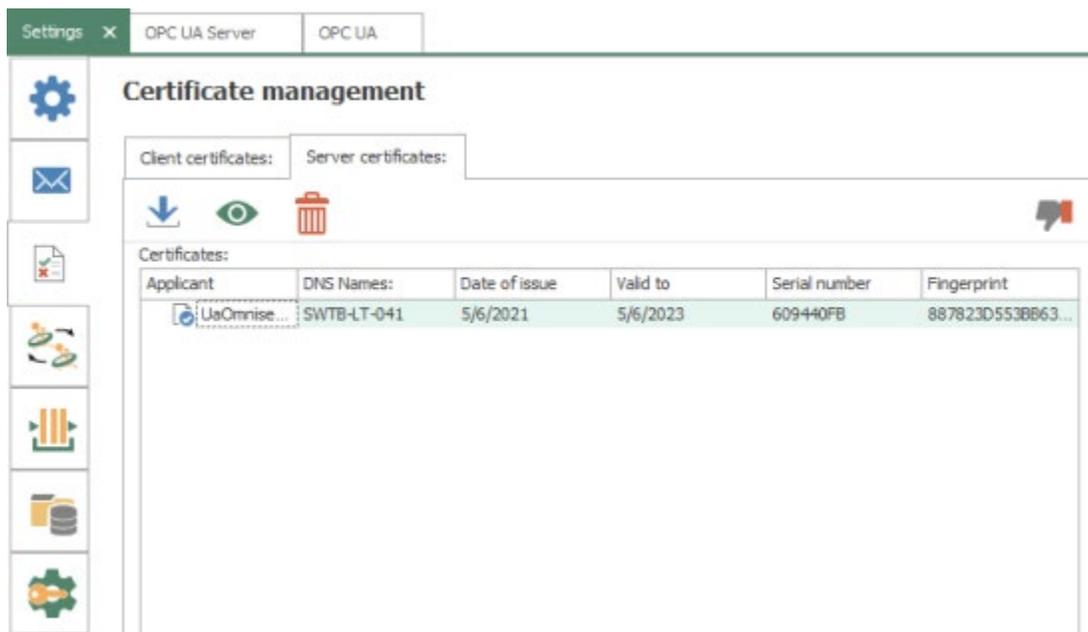
The "Client certificates" tab lists the certificates available for use by OPC Router for both OPC UA client connections to other OPC UA servers and for OPC Router's OPC UA server interface. These are the certificates that will be available from the dropdowns when configuring OPC UA plug-in instances and OPC UA Server plug-in instances.



The following options are available for managing client certificates:

1. **Import** - allows you to import a certificate from a third-party certificate authority such as Verisign or Thawte, etc for use by OPC Router.
2. **Export** - allows you to export the selected certificate (for use in importing into OPC UA clients or servers that will be connecting to OPC Router). Saves as a .crt file or .cer file.
3. **Export with Private Key** - allows you to export the selected certificate with it's private key. Saves as a .pfx file - you will be prompted to assign a password to the file. The private key for a certificate is used for signing purposes.
4. **View** - allows you to view the properties of the selected certificate.
5. **Add Private Key** - for certificates that do not have an existing private key, this option will allow you to add one. For certificates with an existing private key, you will be prompted to overwrite the existing private key.
6. **Delete** - allows you to delete the selected certificate.
7. **Add New** - allows you to create a new self-signed certificate that will be available for use by OPC Router for OPC UA client and server connections.

The "Server certificates" tab manages certificates for external OPC UA clients and servers.



The following options are available for managing server certificates:

1. **Import** - allows you to import the certificate from external OPC UA client and servers (for details on exporting the certificate from your OPC UA client or server, please consult its technical documentation). You can import .pfx, .der, .cer or .crt format certificate files.

2. **View** - allows you to view the properties of the selected certificate.
3. **Delete** - allows you to delete the selected certificate.
4. **Trust/Reject** - allows you to trust a selected imported certificate that is currently not trusted (displayed with a red "X"). For currently trusted certificates (displayed with a blue "check"), this button allows you to reject the certificate such that it is no longer trusted.

## Connecting Your OPC UA Client to OPC Router

Using the information we've just discussed, you can get your OPC UA client connected to OPC Router. And I encourage you to [watch our tutorial video on connecting an OPC UA Client to OPC Router here](#).

## Connecting OPC Router to Another OPC UA Server

Again using the information we discussed earlier, you can get OPC Router connected to your OPC UA server. To that end, I recommend that you [watch this tutorial video that covers connecting OPC Router to other OPC UA servers here](#).

In addition to the tutorial videos for connecting OPC Router with OPC UA clients and servers, another important resource is the [OPC Router Video Resources web page](#). It contains a number of detailed how-to videos on a range of topics related to configuring OPC Router.

## Conclusions

In this e-book, you have learned about the intricacies of security as it pertains to how OPC UA clients and servers connect, as well as some key considerations as you create a secure architecture for your own systems. If you have more questions about OPC UA, a good next step would be reviewing the [many free OPC UA learning resources we have available here](#).

If you have questions about the topics discussed in this e-book or would like to discuss your challenges in designing your own secure OPC UA system, please email us at [whitepapers@softwaretoolbox.com](mailto:whitepapers@softwaretoolbox.com) or call us on +1 888 665 3678 (US/Canada) or +1 704 849 2773 (Global).

This e-book is intended to provide general educational information about OPC UA and security and should not be used as a standalone resource for making any decision about your IT or OT security. Always consult and involve your IT, Cybersecurity team and follow any of your industry specific best practices or those from NIST, CISA and related governmental authorities when making implementation decisions. You are responsible for all decisions made regarding your systems and security.



888 665 3678 TOLL FREE  
+1 704 849 2773 GLOBAL  
Charlotte, NC USA GLOBAL HQ  
[www.softwaretoolbox.com](http://www.softwaretoolbox.com) WEB

Our mission is to provide you with the right software package to solve your industrial operation challenges.